

**БЕЛКООПСОЮЗ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БЕЛОРУССКИЙ ТОРГОВО-ЭКОНОМИЧЕСКИЙ
УНИВЕРСИТЕТ ПОТРЕБИТЕЛЬСКОЙ КООПЕРАЦИИ»**

**С. М. МОВШОВИЧ
В. И. МИСЮТКИН**

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

**Пособие
для студентов специальности 1-26 03 01
«Управление информационными ресурсами»**

Рецензенты: Е. А. Левчук, канд. техн. наук, доцент кафедры
информационно-вычислительных систем
Белорусского торгово-экономического университета
потребительской кооперации;
О. А. Кравченко, канд. физ.-мат. наук, доцент
кафедры информационных технологий Гомельского государственного технического уни-
верситета
им. П. О. Сухого

Рекомендовано к изданию научно-методическим советом учреждения образования «Белорусский торго-
во-экономический университет потребительской кооперации». Протокол № 2 от 11 декабря 2007 г.

М 74 **Мовшович, С. М.**
Технологии программирования : пособие для студентов специаль-
ности 1-26 03 01 «Управление информационными ресурсами» / С. М.
Мовшович, В. И. Мисюткин. – Гомель : учреждение образования «Бело-
русский торгово-экономический университет потребительской коопера-
ции», 2008. – 116 с.
ISBN 978-985-461-613-1

УДК 004.42
ББК 32.973.26-018.2

ISBN 978-985-461-613-1

© Учреждение образования «Белорусский
торгово-экономический университет
потребительской кооперации», 2008

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

При изучении студентами экономических специальностей курсов, связанных с информатикой, ставится задача приобретения будущими специалистами умений и навыков использования информационных технологий в своей предметной области. Приобретенные в вузе умения и навыки использования информационных технологий должны дать возможность будущему специалисту:

- самостоятельно автоматизировать многие виды деятельности на своем рабочем месте с помощью приложений пакета MS Office;
- быстро осваивать применяемые в организации автоматизированные рабочие места специалистов (АРМы), при этом не важно, изучались ли данные АРМы в вузе;
- участвовать в разработке технических заданий на создание новых или модификацию существующих АРМов, комплексов АРМов, автоматизированных информационных систем (АИС).

Отметим важность последней из перечисленных задач подготовки студентов в области информационных технологий. Молодой специалист будет востребован и вовлечен в процессы развития информационных технологий и ресурсов предприятия, если он способен алгоритмизировать бизнес-процессы и владеет системным подходом как в своей предметной области, так и в области информационных технологий.

Курс «Технологии программирования» базируется на курсах «Алгоритмизация и программирование», «Автоматизация экономических расчетов».

Целью курса «Технологии программирования» является развитие у студентов знаний о методах алгоритмизации и формирование умений и навыков программирования в среде *Visual Basic for Application* (VBA).

Задачи данного курса сводятся к получению студентами сведений о системе программирования VBA и выработке практических навыков по созданию приложений на рабочем месте экономиста.

Настоящее пособие содержит следующие разделы:

- Теоретические основы курса.
- Задания лабораторных работ по разработке приложений в среде VBA с примерами и необходимыми теоретическими сведениями.
- Вопросы для подготовки к тестированию и экзамену для студентов заочной формы обучения.
- Примерные тестовые задания.

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КУРСА

1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ О СОЗДАНИИ И ИСПОЛЬЗОВАНИИ МАКРОСОВ

1.1. Макрос: основные понятия

Некоторые действия в Microsoft Excel могут носить повторяющийся характер: дополнение одинаковыми заголовками целого ряда рабочих документов или одним и тем же способом форматирование всех заголовков, периодическое выполнение одних и тех же расчетов. Кроме того, иногда возникает потребность в расширении возможности работы с документами (дополнении интерфейса приложения новыми окнами диалога, панелями инструментов). Во всех случаях можно создать специальную процедуру – *макрос* и затем, запуская его, выполнять такие операции автоматически.

Макрос приложений MS Word, MS Excel, MS PowerPoint, MS Front-Page – это записанная под определенным именем последовательность команд и функций, хранящаяся в модуле VBA. Макрос в указанных приложениях используется для многократного выполнения одной и той же задачи. Для языка VBA макрос – это процедура.

С каждым макросом связываются следующие понятия: *Имя макроса* и *Запись последовательных действий*. Кроме того, каждый макрос для лучшей его идентификации может иметь описание.

Имя макроса – любое сочетание букв и знака подчеркивания. Другие символы не допустимы. Если имя макроса состоит более чем из одного слова, нужно воспользоваться символами подчеркивания (Формат_данных_таблицы, Выделить_все_данные) либо каждое слово начинать с заглавной буквы (ФорматДанныхТаблицы, ВыделитьВсеДанные).

Описание макроса – это краткий текст, поясняющий назначение макроса.

Очевидно, что имя и описание макроса должны соответствовать его назначению.

Записать последовательность действий, или *создать макрос*, можно с помощью следующего:

- автоматической записи макроса;
- ввода текста процедур на макроязыке в редакторе Visual Basic.

Автоматическая запись макроса (или просто запись макроса) – автоматическое протоколирование введенной с клавиатуры информации и всех команд с помощью макроязыка Visual Basic for Applications (VBA) в виде процедур.

Ввод текста процедур на макроязыке осуществляется с помощью встроенного редактора VBA.

И в том, и в другом случае текст макроса размещается в программном модуле или просто модуле.

Отличие двух способов создания макроса очевидно: в первом случае знание макроязыка необязательно. Однако необходимо помнить, что любая команда, даже ошибочная, будет записана в создаваемый макрос, поэтому так или иначе вам придется вносить изменения в свои программы.

1.2. Запись макроса

1.2.1. Этапы записи макроса

Самый простой способ создать процедуру – ее записать. Макрос в этом случае можно рассматривать как простую процедуру *Sub*.

Так как любое нажатие клавиши или щелчок мыши автоматически фиксируется в тексте процедуры, прежде чем записать макрос, необходимо заранее продумать следующее:

1. Условия, которые должны выполняться при запуске макроса:

- местоположение курсора;
- активизация определенной ячейки или листа рабочей книги;
- режим работы приложения;
- открытие определенного файла.

2. Действия, которые должен выполнять макрос. Иногда бывает полезно перед записью макроса потренироваться, чтобы в процедуре было меньше ошибок.

3. Условия завершения работы макроса.

Хороший макрос характеризуется тем, что по завершении своей работы приложение имеет то же состояние, что и до его запуска, конечно, если назначение макроса не состоит в том, чтобы изменить состояние.

Для записи макроса нужно выполнить следующие действия:

- 1) включить процедуру записи макроса;
- 2) определить последовательность команд макроса;
- 3) выключить запись макроса.

Первое и третье действия являются вспомогательными и необходимы для обеспечения записи последовательности команд на макроязыке. Очевидно, что для разных макросов последовательность команд различна. Вспомогательные же действия одинаковы для всех макросов.

Включение процедуры записи макроса осуществляется при выполнении команды *Сервис\Макрос\Начать запись*. Появляется диалоговое окно *Запись макроса* (рис. 1).

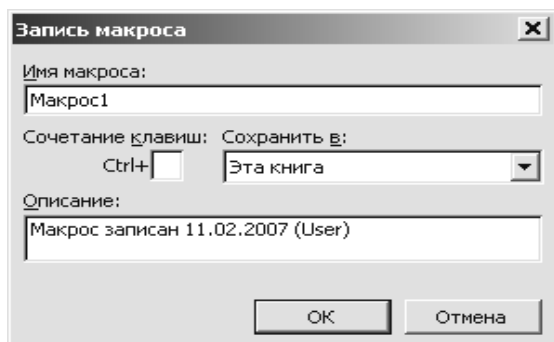


Рис. 1. Диалоговое окно *Запись макроса*

В поле ввода *Имя макроса* вводится имя макроса, а в поле ввода *Описание* – краткий комментарий о назначении макроса. После нажатия кнопки *ОК* на рабочем листе Excel появляется небольшая панель *Остановка записи* (рис. 2) с тремя кнопками: *Остановить запись*, *Относительная ссылка* и *Заккрыть*.

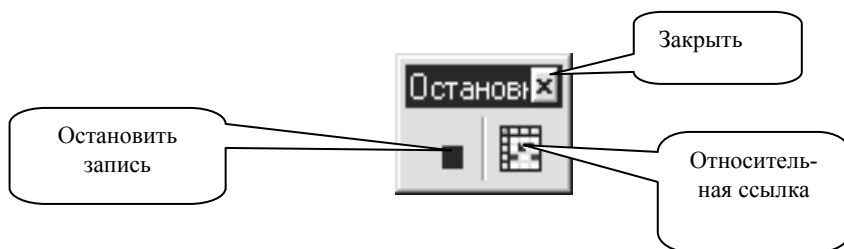


Рис. 2. Панель *Остановка записи*

Выключить запись макроса можно двумя способами:

- 1) с помощью кнопки *Остановить запись*;
- 2) командой *Сервис\Макрос\Остановить запись*.

Записывать последовательность команд в макрос можно с использованием абсолютных и относительных ссылок на ячейки рабочего листа. В первом случае, независимо от того, какая ячейка на момент выполнения макроса является активной, действия будут выполняться над конкретными ячейками рабочего листа. В случае записи макроса с относительными ссылками работа осуществляется каждый раз с разными ячейками, если на момент запуска макроса активными были разные ячейки.

1.2.2. Запись макроса с абсолютными ссылками

Макрос с абсолютными ссылками предназначен для ввода данных в конкретные ячейки рабочего листа.

Для записи макроса необходимо выполнить следующие действия:

- 1) активизировать любую ячейку рабочего листа (ввод данных начнется именно с нее);
- 2) выполнить команду *Сервис\Макрос\Начать запись*;
- 3) ввести в поле ввода *Имя макроса* диалогового окна название создаваемого макроса, а в поле *Описание* – краткое описание назначения макроса;
- 4) нажать кнопку *ОК*;
- 5) выполнить на рабочем листе Excel последовательность действий, которая будет записана в макрос;
- 6) нажать кнопку *Остановить запись*.

Для запуска записанного макроса на выполнение активизируйте команду *Сервис\Макрос\Макросы*. Появится диалоговое окно *Макрос*. В списке имен макросов выберите имя созданного макроса и нажмите кнопку *Выполнить*.

Чтобы убедиться в правильной работе макроса, очистите рабочий лист и активизируйте ранее не использовавшуюся ячейку. Выполните макрос. Результатом работы будет заполнение тех же ячеек, которые были указаны при создании макроса. Итак, результат выполнения макроса не зависит от того, где находился курсор перед его запуском.

1.2.3. Запись макроса с относительными ссылками

Очень часто необходимо работать не с конкретными, а с любыми ячейками или областью рабочего листа. Для этого при записи макроса необходимо включить режим работы с относительными ссылками.

В этом случае надо выполнить следующие действия:

- 1) активизировать любую ячейку рабочего листа;
- 2) выполнить команду *Сервис\Макрос\Начать запись*;
- 3) ввести в поле ввода *Имя макроса* диалогового окна название создаваемого макроса, в поле *Описание* ввести краткое описание назначения макроса;
- 4) нажать кнопку *ОК*;
- 5) на панели *Остановка записи* нажать кнопку *Относительная ссылка*;
- 6) выполнить на рабочем листе Excel последовательность действий, которая будет записана в макрос;
- 7) нажать кнопку *Остановить запись*.

Убедитесь в правильности работы макроса. Для этого активизируйте, например, ячейку В3. Выполните макрос. Результатом работы макроса будет заполнение (форматирование) ячеек, начиная с В3.

Таким образом, отличие записи макроса с абсолютными и относительными ссылками заключается только в том, что перед вводом команд необходимо включить режим работы с относительными ссылками, нажав кнопку *Относительная ссылка* на панели *Остановка записи*.

Иногда возникает такая ситуация: с помощью кнопки *Закрыть* панели *Остановка записи* панель была удалена с рабочего листа. Причем при последующих обращениях к команде *Начать запись* она уже не появляется, а необходимо записать макрос, работающий с относительными ссылками. В этом случае необходимо выполнить следующие действия:

- 1) выполнить команду *Вид\Панели инструментов\Настройка*, после чего появится диалоговое окно *Настройка*;
- 2) выбрать вкладку *Панели инструментов*;
- 3) установить флажок *Остановка записи*;
- 4) закрыть окно нажатием кнопки *Создать*.

1.3. Удаление макроса

Для удаления макроса необходимо выполнить следующие действия:

- 1) выполнить команду *Сервис\Макрос\Макросы*, в результате чего появится диалоговое окно *Макрос*;
- 2) выделить в списке макросов имя удаляемого макроса;
- 3) нажать кнопку *Удалить*.

1.4. Редактирование макроса

Часто приходится редактировать записанные макросы с целью изменения или добавления некоторых команд. После активизации редактора VBA редактирование можно выполнить следующим образом:

- 1) выполнить команду *Сервис\Макрос\Макросы*, в результате чего появится диалоговое окно *Макрос*;
- 2) выделить в списке макросов имя требуемого макроса;
- 3) нажать кнопку *Изменить*, после чего появится окно модуля с текстом макроса;
- 4) внести необходимые изменения в текст процедуры;
- 5) для выхода из редактора на рабочий лист Excel нажать на расположенную слева на панели инструментов кнопку пиктограммы Excel.

1.5. Переименование макроса

Специальной команды для переименования макроса нет. Однако часто возникает потребность исправить имя или присвоить макросу совершенно новое имя. Для этого нужно войти в режим редактирования макроса и в тексте процедуры исправить заголовок. Новое имя автоматически заменит старое в списках макросов.

1.6. Вызов макроса или запуск макроса на выполнение

Вызвать макрос или запустить его на выполнение можно несколькими способами:

- командой основного меню *Сервис\Макрос\Макросы\Выполнить*;
- клавишами быстрого вызова;
- активизацией графического объекта;
- кнопкой.

Запуск макроса на выполнение с помощью команды основного меню *Сервис\Макрос\Макросы\Выполнить* является самым простым.

Клавиши быстрого вызова удобно использовать для часто выполняемых макросов.

Графические объекты и кнопки используются для создания удобного и красивого диалога.

1.7. Назначение макросу клавиш быстрого реагирования

Назначение макросу клавиш быстрого реагирования может осуществляться непосредственно на стадии создания макроса, а также после создания макроса.

В первом случае в диалоговом окне *Запись макроса* (см. рис. 1) в поле *Сочетание клавиш* необходимо ввести букву, одноименная клавиша которой в сочетании с клавишей *Ctrl* и будет использоваться для быстрого вызова макроса. Клавишам быстрого вызова назначаются только буквы латинского алфавита.

Для назначения уже созданному макросу клавиш быстрого вызова необходимо выполнить следующее:

- активизировать команду *Сервис\Макрос\Макросы*, в результате появится диалоговое окно *Макрос*;
- выделить имя макроса;
- нажать кнопку *Параметры*, после чего появится диалоговое окно *Параметры макроса*;
- выполнить назначение клавиш.

1.8. Назначение макроса графическому объекту

Макрос может вызываться нажатием кнопки мыши на графическом объекте.

Графический объект – это созданное пользователем любое графическое изображение и рисунок, в том числе и из библиотеки MS Office.

Для назначения макроса графическому объекту необходимо сначала создать макрос и поместить графический объект на рабочем листе Excel. Два этих процесса (создание макроса и графического объекта) могут выполняться в произвольном порядке.

Создание графического объекта осуществляется с помощью панели инструментов *Рисование* или с помощью команды *Вставка\Рисунок*.

Назначение макроса графическому объекту осуществляется достаточно просто. Для этого необходимо щелкнуть кнопкой на графическом объекте для его выделения, а затем нажатием правой кнопки мыши вызвать контекстное меню. В нем следует выбрать команду *Назначить макросу*. В появившемся диалоговом окне нужно выделить имя макроса и закрыть окно.

1.9. Назначение макроса кнопке

По способу расположения кнопок различают стандартные кнопки Excel, расположенные на рабочем листе, и кнопки панели инструментов.

Как и графические объекты, стандартные кнопки располагаются на рабочем листе Excel. Кнопки пане-

ли инструментов могут располагаться только в области панелей инструментов среди кнопок уже имеющих на экране панелей, например, *Стандартная*, *Форматирования*, или на специально созданной панели пользователя.

И в случае использования стандартной кнопки для запуска макроса, и в случае использования кнопки панели управления макрос должен быть создан раньше, чем кнопка. Если не выполнить это требование, возникают проблемы с назначением макроса кнопке.

1.9.1. Назначение макроса стандартной кнопке

Для назначения макроса стандартной кнопке необходимо выполнить следующее:

- 1) создать макрос;
- 2) создать на рабочем листе стандартную кнопку и назначить ее макросу:
 - выполнить команду *Вид\Панели инструментов*, отметить щелчком левой кнопки мыши панель *Формы*, после чего на рабочем листе появится панель с элементами управления (рис. 3);

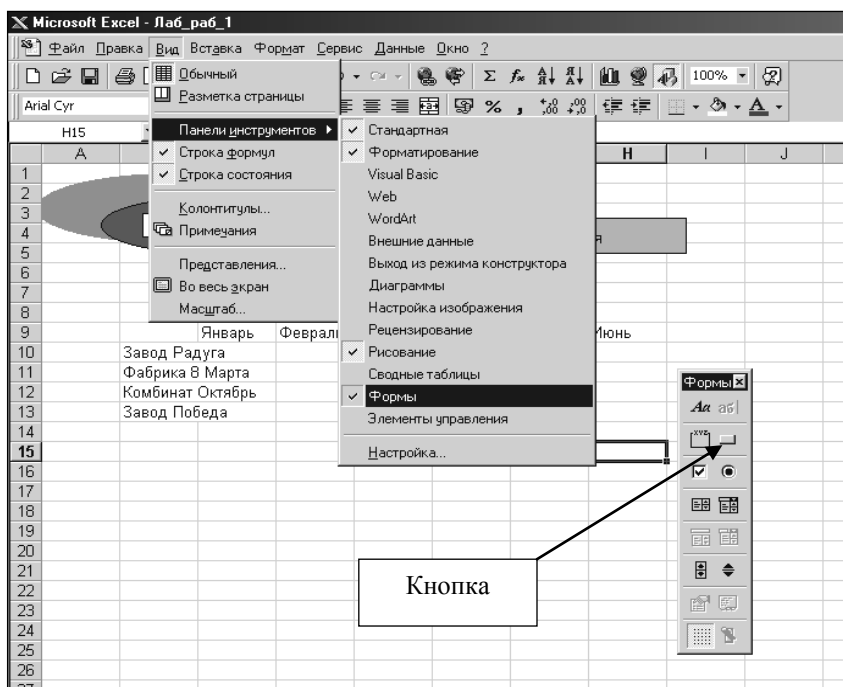


Рис. 3. Вывод на рабочий лист панели *Формы*

- щелчком левой кнопки мыши на панели *Формы* выбрать элемент *Кнопка* и на рабочем листе с помощью мыши определить его размер;
 - в диалоговом окне *Назначить макрос объекту* необходимо выбрать имя нужного макроса;
 - можно изменить название созданной кнопки, щелкнув на ней левой кнопкой мыши;
 - щелкнуть кнопкой мыши в любой области рабочего листа.
- Создание стандартной кнопки и назначение ей макроса завершено.

1.9.2. Назначение макроса кнопке панели управления

Для назначения макроса кнопке панели управления необходимо выполнить следующее:

- 1) создать макрос;
- 2) создать пользовательскую кнопку панели инструментов следующим образом:
 - выполнить команду *Сервис\Настройка*, в диалоговом окне *Настройка* выбрать вкладку *Команды*;
 - в списке *Категории* выделить строку *Макросы*, в списке *Команды* щелкнуть на кнопке *Настраиваемая кнопка* и перетащить ее в область панелей инструментов (крайняя правая кнопка в нижнем ряду панелей инструментов на рис. 4);
- 3) создать свой рисунок на данной кнопке (не закрывая окно *Настройка*):
 - щелкнуть правой кнопкой мыши по кнопке, которая выведена на панель инструментов, после чего появится контекстное меню (рис. 4);
 - выбрать команду *Изменить значок на кнопке*, в результате чего появится *Редактор рисунков* (рис. 5);
 - нажать кнопку *Очистить*;
 - выбрать в окне *Цвета* нужный для изображения цвет и, перемещая курсор мыши с нажатой левой кнопкой мыши по рисунку кнопки, нарисовать требуемое изображение; по завершении редактирования рисунка нажать кнопку *OK*;

- 4) назначить кнопке макрос и присвоить ей имя:
- при открытом окне *Настройка* щелкнуть правой кнопкой мыши по созданной кнопке, в результате чего появится контекстное меню (рис. 4), выбрать команду *Назначить макрос*;
 - завершить назначение выбором макроса из списка макросов; закрыть окно *Настройка*;
 - для присваивания пользовательской кнопке панели инструментов своего имени надо активизировать контекстное меню для созданной кнопки (рис. 4), в поле *Имя* ввести имя созданной кнопки (при помещении указателя мыши на эту кнопку будет появляться текстовое поле с именем кнопки).

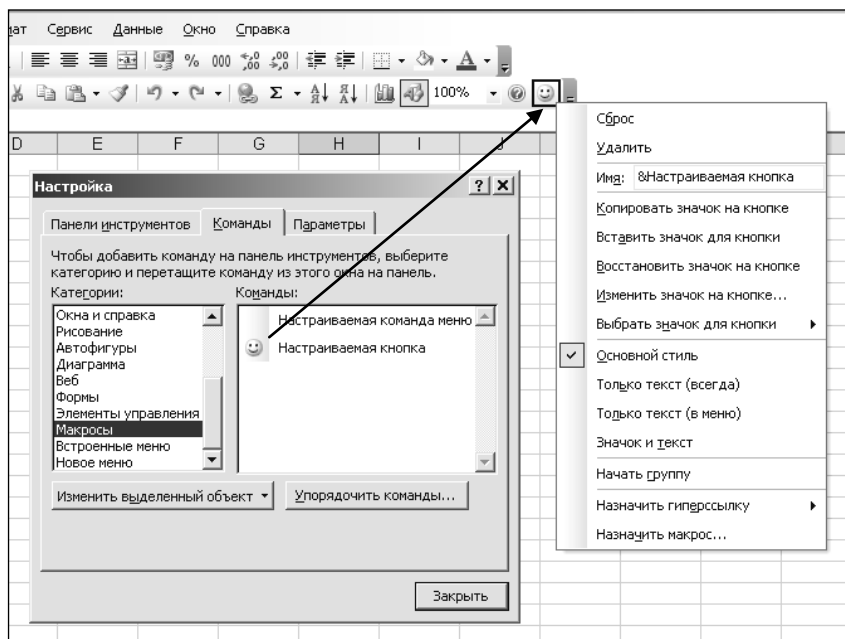


Рис. 4. Назначение пользовательской кнопке панели управления имени и макроса



Рис.5. Окно Редактора рисунков

1.10. Создание панели управления пользователя

Для создания панели управления пользователя необходимо выполнить следующие действия:

- 1) создать столько макросов, сколько кнопок будет размещено на панели инструментов пользователя;
- 2) присвоить создаваемой панели собственное имя:
 - активизировать команду *Вид\Панели инструментов\Настройка*, в результате чего появится диалоговое окно *Настройка*;
 - выбрать вкладку *Панели инструментов* и нажать кнопку *Создать*, после чего появится окно *Создание панели инструментов* (рис. 6);

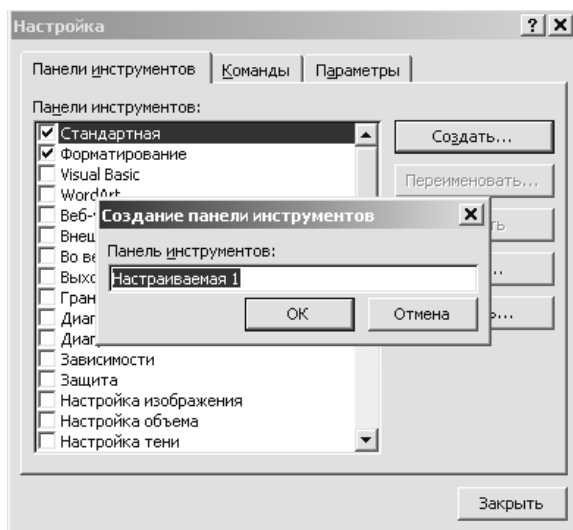


Рис. 6. Присваивание создаваемой панели инструментов пользователю собственного имени

- в поле ввода *Панель инструментов* ввести имя, которое вы хотите присвоить создаваемой панели инструментов;
- закрыть окно, нажав кнопку *ОК* (на рабочем листе появится новая панель, но она еще не содержит ни одной кнопки);
- 3) разместить кнопки на созданной панели инструментов:
 - выбрать вкладку *Команды* в диалоговом окне *Настройка*;
 - выделить в списке *Категории* строку *Макросы*, а в списке *Команды* – *Настраиваемая кнопка*;
 - перетащить изображение настраиваемой кнопки в область создаваемой панели инструментов пользователя; щелкнуть правой кнопкой мыши по кнопке пользователя на создаваемой панели управления и (или) в окне *Настройка* по кнопке *Изменить выделенный объект*, вызвать контекстное меню (рис. 7) и выбрать команду *Выбрать значок для кнопки* (появится окно с набором кнопок), выделить любое понравившееся изображение и щелкнуть по нему левой кнопкой мыши (изображение кнопки сменится);

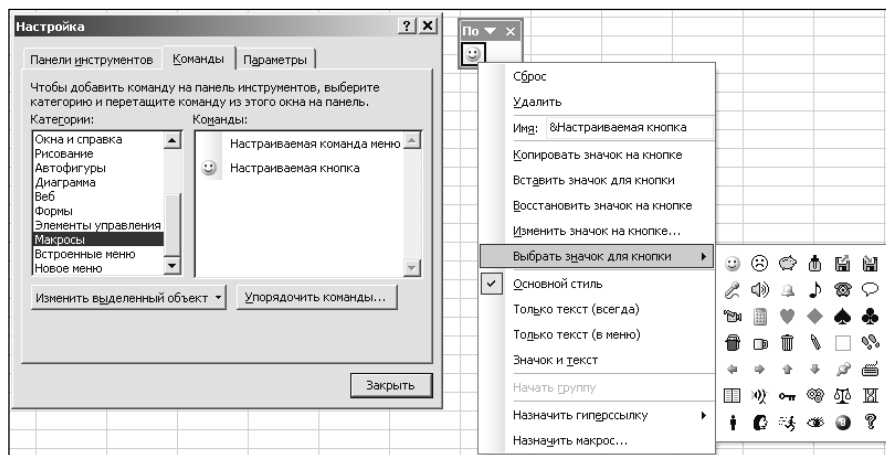


Рис. 7. Размещение кнопок на панели инструментов пользователя

- с помощью команды *Назначить макрос* поставить в соответствие кнопке нужный макрос;
- в поле *Имя* контекстного меню ввести имя кнопки;
- перетащить необходимое количество настраиваемых кнопок из списка *Команды* окна *Настройка* в область создаваемой панели инструментов;
- изменить изображения на кнопках, назначить им макросы и присвоить имена;
- перетащить созданную панель с рабочего листа в область панелей инструментов;
- в списке панелей инструментов появится новая запись – имя созданной вами панели инструментов.

Если созданная панель управления уже не нужна, ее можно удалить, а для этого – выделить и нажать кнопку *Удалить*.

Описанная технология создания пользовательской панели может быть использована не только для вызова макросов, но и для команд, имеющихся в списке команд и не попавших в имеющиеся панели инструментов.

2. ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ В ТАБЛИЧНОМ ПРОЦЕССОРЕ EXCEL

2.1. Правила набора пользовательских функций

Для того чтобы попасть в редактор VBA, надо выполнить команду *Сервис\Макрос\Редактор Visual Basic*. В результате откроется два окна: *Project-VBA Project* и *Properties*.

В окне *Project-VBA Project* отображается реестр модулей и форм, входящих в создаваемый проект. Модуль – это лист, в котором набирается код. Точнее, в редакторе VBA у каждого рабочего листа имеется собственный модуль. Указанные листы не надо путать с рабочими листами книги Excel. Рабочая книга обладает своим листом. Кроме того, если в проекте есть пользовательские формы, то каждая из них имеет свой модуль.

Двойным щелчком на значке модуля в окне *Project-VBA Project* можно открыть соответствующий модуль. Значок активного модуля выделяется серым цветом.

Для создания пользовательской функции надо добавить в проект стандартный модуль. Добавление стандартного модуля выполняется по команде *Insert\Module*. В результате появится окно листа модуля, приведенное на рис. 8.

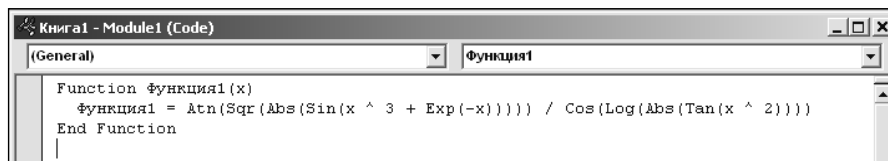


Рис. 8. Пример кода математической пользовательской функции

На листе набирается текст (код) пользовательской функции. В общем случае пользовательская функция имеет следующий вид:

Function имя функции (список аргументов)

тело функции

End Function

Тело функции представляет последовательность операторов, среди которых должен быть, по крайней мере, один оператор присваивания, в левой части которого указано имя функции. В этом операторе имени функции присваивается результат.

Список аргументов представляет собой последовательность имен переменных, разделенных запятыми.

При правильном наборе первой строки последняя строка появляется автоматически.

2.2. Математические пользовательские функции

В качестве примера рассмотрим создание пользовательской функции для вычисления выражения

$$y = \frac{\arctg \sqrt{|\sin(x^3 + e^{-x})|}}{\cos(\ln |tg x^2|)}.$$

В стандартном модуле надо набрать код, приведенный на рис. 8.

В данной функции были использованы встроенные функции. Список встроенных математических функций приведен в табл. 1.

Таблица 1. Встроенные математические функции VBA

Математическая запись	Запись на языке VBA	Примечание
$ x $	Abs(x)	Абсолютная величина числа
$\arctg x$	Atn(x)	Арктангенс
$\cos x$	Cos(x)	Косинус
e^x	Exp(x)	Экспонента
$\ln x$	Log(x)	Натуральный логарифм
$\text{sign } x$	Sgn(x)	Возвращает знак числа: 1, 0 или -1
$\sin x$	Sin(x)	Синус
\sqrt{x}	Sqr(x)	Квадратный корень
$\text{tg } x$	Tan(x)	Тангенс
$[x]$	Fix(x)	Если $x > 0$, то функция Fix возвращает целое число, ближайшее меньшее к x Если $x < 0$, то функция Fix возвращает целое число, большее или равное x
$[x]$	Int(x)	Для любого x функция Int возвращает целое число, ближайшее меньшее к x

Использование созданной функции происходит так же, как и любой встроенной функции MS Excel. После создания функции ее можно найти в окне мастера функций MS Excel в списке категории *Определенные пользователем*.

После выбора функции из списка пользовательских функций откроется ее окно, в котором надо сделать ссылки на ячейки рабочего листа со значениями аргументов.

2.3. Разветвляющиеся пользовательские функции

Как известно, для реализации разветвляющейся функции необходим условный оператор. Основная форма условного оператора в языке VBA похожа на условный оператор языка Паскаль, но в отличие от последнего в каждой части оператора можно записать не один, а несколько операторов:

- 1) *if условие then*
 операторы1
 else операторы2

Так же, как и в Паскале существует сокращенная форма без *else*:

- 2) *if условие then операторы*

Однако в языке VBA есть еще блочная форма записи условного оператора, которая упрощает чтение программы:

- 3) *if условие then*
 операторы1
 else операторы2
 end if

Для записи вложенных условных операторов используется следующая форма:

- 4) *if условие1 then*
 операторы1
 elseif условие2 then
 операторы2
 else операторы3
 end if

На рис. 9 приведен пример записи следующей функции:

$$y = \begin{cases} x^2, & \text{если } x < 0; \\ 2 \cdot x, & \text{если } x > 0; \\ 10, & \text{если } x = 0. \end{cases}$$

Для создания этой функции необходимо использовать два вложенных друг в друга условных оператора. Обратите внимание, что слово *then* пишется в той же строке, что и *if*. В противном случае выдается сообщение об ошибке.

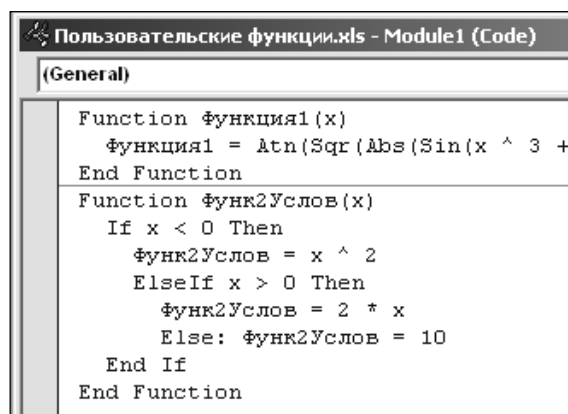


Рис. 9. Пример функции с проверками двух условий

3. ОСНОВЫ ПРОГРАММИРОВАНИЯ В СРЕДЕ VBA

3.1. Последовательность набора программ в среде VBA

Набор программ в среде VBA осуществляется в следующей последовательности:

1. *Запуск* табличного процессора *Excel*.

2. *Открытие* существующей рабочей книги или *создание новой* (новая книга создается при запуске Excel). Все созданные процедуры будут сохраняться вместе с рабочей книгой в одном файле.

3. *Запуск редактора Visual Basic (VB)* производится с помощью команды *Сервис\Макрос\Редактор Visual Basic* или комбинации клавиш *Alt+F11*.

4. *Ввод текста процедуры* в окно программы:

4.1. При создании первой программы в проекте или при создании нового модуля необходимо выполнить следующее:

- выделить нужный проект в окне проекта;
- добавить модуль к проекту при помощи команды *Insert\Module* или использовать соответствующую кнопку на панели инструментов *Стандартная*;
- ввести текст новой процедуры в окне программы.

4.2. Для загрузки существующей процедуры или ввода новой в существующий модуль нужно сделать следующее:

- открыть окно программы для модуля;
- выполнить двойной щелчок по имени модуля в окне проекта или выделить модуль в окне проекта и выполнить команду *View\Code (F7)*;
- ввести текст новой процедуры или изменить текст.

При вводе текста процедуры или ее изменении редактор VBA определяет ошибки синтаксиса в каждой строке после того, как курсор из нее будет перемещен в другую строку. При этом редактор VBA выполняет синтаксический анализ строки и ее компиляцию. Если при этом будут обнаружены ошибки, то вся строка выделяется красным цветом, а на экран выводится диалоговое окно с сообщением об ошибке и кнопками *Справка* и *ОК*. Для доступа к справочной системе VBA и получения сведений об ошибке щелкните по кнопке *Справка*, для удаления с экрана диалогового окна щелкните по кнопке *ОК*. Для получения справки по конкретному ключевому слову нужно поместить курсор на это слово и нажать клавишу *F1*. Если ошибок синтаксиса не будет обнаружено, то выполняется цветовое оформление строки (ключевые слова отображаются синим цветом, комментарии – зеленым, остальной текст – черным) и компиляция строки. Рекомендуется сразу же исправлять обнаруженные ошибки синтаксиса до ввода или редактирования следующей строки или до перехода к следующему этапу работы.

5. *Сохранение проекта* выполняется по технологии сохранения рабочей книги Excel.

6. *Выполнение (запуск) процедуры*:

- установить курсор на текст нужной процедуры;
- выполнить команду *Run\Run Sub\UserForm (F5)* или использовать соответствующую кнопку на панели инструментов *Стандартная*.

При выполнении процедур редактор VBA выявляет ошибки, которые не могли быть обнаружены при вводе и редактировании программ. Такие ошибки называются ошибками времени исполнения (*runtime errors*), или *runtime*-ошибками. При обнаружении таких ошибок выполнение процедуры прекращается и на экран выводится диалоговое окно с сообщением об ошибке и кнопками *Continue* (продолжить), *End* (завершить), *Debug* (отладка) и *Help* (помощь). Для большинства ошибок кнопка *Continue* отключена, *End* завершает работу процедуры. Кнопка *Debug* позволяет перейти в режим отладки или продолжить работу процедуры (например, повторить ввод данных), при этом команда и кнопка *Run Sub\UserForm* превращаются соответственно в команду и кнопку *Continue* (продолжить).

Для прерывания работы процедуры (например, для выхода из закликивающейся программы) можно также использовать команды меню *Run* и соответствующие кнопки на панели инструментов *Стандартная*:

- *Break* – для прерывания выполнения процедуры и перехода в режим отладки;
- *Reset* – для установления процедуры в исходное состояние, выхода из режима отладки.

7. *Ввод исходных данных, анализ результатов* и подготовка их для вывода на принтер.

При выполнении процедур исходные данные (если они есть) обычно вводятся в режиме диалога в текстовое окно ввода, а результаты могут быть размещены в ячейках рабочего листа, в текстовом файле или выведены в диалоговое окно (которое имеет, по крайней мере, одну кнопку *ОК*). Если результаты выведены в диалоговое окно, то после их анализа необходимо нажать кнопку *ОК* и вернуться в окно редактора.

Если редактор VBA обнаружил ошибки времени исполнения или полученные результаты не совпадают с подготовленными тестами, то следует перейти к редактированию процедуры.

8. *Редактирование процедур*.

При редактировании процедур используются методы работы с текстом в любых текстовых редакторах в операционной системе Windows. Для работы с буфером обмена, поиска фрагментов текста можно использовать команды меню *Edit* и соответствующие кнопки на панели инструментов *Стандартная*. Для каждой редактируемой строки редактор VBA определяет ошибки синтаксиса.

9. *Закрытие окна редактора* и переход в Excel.

Необходимо выполнить команду *File\Close and Return to Microsoft Excel* или нажать комбинацию клавиш *ALT+Q*.

10. *Завершение работы Excel*.

Необходимо закрыть окно Excel.

3.2. Структура простой программы

Программа, состоящая из одной процедуры, имеет следующий вид:

Sub имя процедуры(список аргументов)

последовательность инструкций

End Sub

Список аргументов может быть пустым.

Последовательность инструкций – это последовательность описаний и операторов.

Среди операторов могут быть несколько операторов *Exit Sub*. Этот оператор обеспечивает выход из процедуры, т. е. прекращение ее работы. Если таких операторов нет в «теле процедуры», то процедура прекращает работу при достижении конца процедуры *End Sub*.

3.3. Допустимые имена (идентификаторы)

Идентификаторы пользователя применяются для обозначения переменных, констант, процедур, функций, различных объектов, определенных самим пользователем.

При выборе имен идентификаторов необходимо руководствоваться следующими правилами:

- длина имени не должна превышать 255 символов;
- имя может содержать любую комбинацию букв, цифр и допустимых символов, начинающуюся только с буквы;
- допустимым символом является знак подчеркивания, который может быть размещен в любом месте идентификатора, кроме первого;
- имена должны быть уникальны внутри области, в которой они определены;
- имена не должны совпадать со служебными словами языка VBA и именами встроенных функций и процедур.

Пример

Допустимые имена: *X15, F_I_O, NomerTel.*

Недопустимые имена: *15x, Ф.И.О., №Tel.*

Идентификаторы языка VBA не чувствительны к состоянию регистра: написание идентификатора прописными или заглавными буквами не имеет значения. Например, идентификаторы пользователя *Name* и *name* или *Sin* и *sin* для редактора VBA представляют собой одно и то же.

3.4. Типы данных

Понятие типа является одним из фундаментальных понятий любого языка программирования. При решении любой задачи требуются данные. Каждый элемент данных или объект (константа, переменная, выражение, функция), которым оперирует программа, относится к определенному типу.

Тип данных определяет:

- 1) множество значений, которые могут принимать объекты программ;
- 2) совокупность операций, допустимых над этими объектами;
- 3) объем выделяемой памяти и форму представления данных в ней.

В редакторе VBA можно обрабатывать числа, строки, логические значения, даты и объекты, использовать наиболее общий тип данных – тип *Variant*, а также создавать собственные типы данных (*Type*). Наиболее часто используемые типы данных представлены в табл. 2.

Таблица 2. Основные типы данных языка VBA

Тип данных	Описание	Размер, байт	Диапазон
<i>Byte</i>	Байт	1	От 0 до 255
<i>Boolean</i>	Логический	2	<i>True</i> или <i>False</i>
<i>Integer</i>	Целый	2	От -32 768 до 32 767 (от -2^{15} до $2^{15}-1$)
<i>Long</i>	Тип «Длинное целое»	4	От -2 147 473 648 до 2 147 483 647 (от -2^{31} до $2^{31}-1$)
<i>Single</i>	Вещественный с плавающей точкой обычной точности	4	От -3,402823E38 до -1,401298E-45 для отрицательных чисел От 1,401298E-45 до 3,402823E38 для положительных чисел
<i>Currency</i>	Денежный	8	От -922337230685477,5808 до 922337230685477,5807
<i>Date</i>	Тип «Дата и время»	8	С 1 января 100 года до 31 декабря 9999 года
<i>String</i>	Строковой переменной длины	256	
<i>Variant</i>	Числовые подтипы	16 байт + N байт (согласно типу)	Переменная любого числового типа

Тип *Variant*, использованный для описания некоторой переменной, позволяет присваивать и обрабатывать с ее помощью данные разных типов. С одной стороны это хорошо: не надо помнить об ограничениях на множество допустимых значений и операций. Но это чревато ошибками, значительно увеличивает время обработки переменных и необходимые для выполнения программы ресурсы памяти.

3.5. Описание переменных

Переменными называются элементы данных, которые при выполнении программы могут принимать различные значения в соответствии с указанным типом.

Для объявления переменных обычно используется инструкция объявления *Dim*.

Синтаксис:

Dim имя переменной [As тип] [, имя переменной [As тип] ...

В случае отсутствия параметра *As тип* переменная будет иметь тип *Variant*.

Например, инструкции

Dim A As Integer

Dim B As Single

Dim C As Boolean

описывают переменные целого (*A*), вещественного (*B*) и логического (*C*) типов.

Это же объявление переменных можно выполнить с помощью одной инструкции *Dim*:

Dim A As Integer, B As Single, C As Boolean

При объявлении любой переменной редактор VBA выполняет ряд действий:

- в памяти выделяется область, размер которой определяется в соответствии с типом объявляемой переменной;
- выделенная область памяти связывается с идентификатором объявляемой переменной, при обращении к переменной фактически происходит обращение к области памяти, где хранится текущее значение переменной;
- переменная инициализируется, т. е. в соответствии с типом объявляемой переменной ей присваивается начальное значение. Числа инициализируются значением 0, переменные *Boolean* – значением *False*, строки – пустыми строками.

Объявить переменную можно в любом месте программы, но обязательно до ее использования. Однако правила хорошего стиля программирования все-таки предписывают делать это в самом начале программы в ее так называемой описательной части.

Переменные, описанные с помощью инструкции *Dim*, называются явно описанными.

Чтобы избежать части ошибок, редактор VBA позволяет наложить требование на явное описание всех переменных в модуле. В этом случае при использовании необъявленной переменной во время компиляции генерируется ошибка, которую можно легко исправить.

Чтобы наложить требование на явное описание переменных, необходимо в разделе описаний модуля указать директиву компилятора *Option Explicit*.

3.6. Операции, операторы и выражения языка VBA

В программах на языке VBA можно использовать стандартный набор операций над данными.

Все операции языка VBA можно разделить на следующие группы:

- арифметические;
- отношения;
- конкатенации (сцепления);
- логические.

Им соответствуют группы операторов с аналогичными названиями.

Арифметические операторы используются для выполнения математических вычислений: $a+b$, $a-b$, $-a$, $a*b$, a/b , $a\b b$ (целочисленное деление), a^b , $a \text{ Mod } b$ (остаток от целочисленного деления).

Оператор конкатенации (&) используется для объединения строковых значений. С его помощью можно формировать строки, состоящие из компонент различного типа. Например:

$y = \text{"Сегодня"} \& 1 \& \text{"Мая"}$

Операторы сравнения ($=$, $<$, $>$, $<=$, $>=$, $<>$) позволяют сравнить два выражения. Они используются при написании простых логических выражений. Результатом сравнения может быть значение *True* (истина), *False* (ложь).

Результатом выполнения *логических операторов* (*and*, *or*, *not*, *xor*) также является логическое значение *True*(истина) или *False*(ложь). Однако логические операторы используются при написании сложных логических выражений. Например:

$i \leq N \text{ and } W$

$(z + 1) <> (x > 3) \text{ and } (y < 5)$

$a > 0 \text{ or } (b < 0 \text{ and } c = 0) \text{ or } g <> a + b$

Выражение – это комбинация операндов (констант, переменных, обращений к функциям), круглых скобок и знаков операций (операторов).

В зависимости от возвращаемого результата и используемых в выражении операторов различают *числовые, строковые, логические* выражения, выражения типа *Дата и время* и *объектные* выражения.

Если выражение содержит несколько операторов (знаков операций), то они выполняются в определенном порядке. Порядок выполнения операций называют *порядком старшинства*, или *приоритетом* операций. Приоритеты выполнения операций приведены в табл. 3.

Таблица 3. Приоритеты операций

Приоритет	Оператор (операция)
1	Вызов функции и скобки
2	^ (возведение в степень)
3	- (смена знака)
4	*, / (умножение, деление)
5	\ (целочисленное деление)
6	Mod (остаток от целочисленного деления)
7	+, – (сложение, вычитание)
8	&, + (конкатенация, слияние строк)
9	=, <>, <, >, <=, >= (операции отношения)
10	Not (логическое отрицание)
11	And (логическое И)
12	Or (логическое ИЛИ)
13	Xor (исключающее ИЛИ)

Стоящие рядом в выражении операторы умножения и деления, сложения и вычитания, а также операторы сравнения выполняются слева направо. Операторы внутри круглых скобок всегда выполняются раньше, чем операторы вне скобок. Порядок выполнения операторов, стоящих внутри скобок, определяется приоритетом операций.

3.7. Оператор присваивания

Оператор присваивания осуществляет присваивание результата вычисления некоторого выражения переменной или константе.

Синтаксис:

<идентификатор> = <выражение>

Элементы синтаксиса:

- *идентификатор* – идентификатор переменной, константы или свойства объекта, удовлетворяющий стандартным правилам именования;
- *выражение* – любое выражение, удовлетворяющее требованиям его составления.

Оператор присваивания предписывает сначала вычислить значение выражения, заданное в его правой части, а затем присвоить полученный результат идентификатору, имя которого указано в левой части оператора.

Ограничений на тип идентификатора и выражения в операторе присваивания нет, т. е. идентификатору может быть присвоено значение любого выражения, даже если их типы не совпадают. Исключением является то, что значение строкового выражения нельзя присвоить идентификатору числового типа.

3.8. Правила оформления кода

VBA не накладывает на структуру программы (процедуры) каких-либо особых ограничений. Однако желательно придерживаться следующих рекомендаций:

1. В соответствии с правилами хорошего стиля программирования надо размещать инструкции описания в начале процедуры, т. е. структурно выделять в ней описательную и исполнительную части.
2. В разделе описаний модуля необходимо указать директиву компилятора *Option Explicit*.
3. Необходимо использовать комментарии. Комментарии, т. е. пояснения к фрагменту текста процедуры, не являются программным кодом и поэтому компилятором игнорируются. Для ввода комментариев используется символ апострофа. Его можно использовать в любом месте строки. При этом все символы, начиная от апострофа до конца строки, будут восприниматься компилятором как комментарий. Комментарий может быть также записан после служебного слова *Rem*.
4. Длинные инструкции целесообразно размещать в нескольких строках. Для этого используют признак продолжения строки, состоящий из двух символов: пробела и подчеркивания.

5. Можно располагать нескольких операторов в одной строке с использованием в качестве разделителя двоеточия. Это всегда короткие операторы, как правило, операторы присваивания.

6. Следует выделять логические уровни кода отступами (записывать текст кода «лесенкой»). Любая информация воспринимается лучше, если она имеет иерархическую структуру (процедура, цикл, вложенный цикл, ветвление, вложенное ветвление, описательные инструкции и т. д.). Правила отступа: основной текст программы следует набирать с отступом, например, в два пробела, каждый вложенный блок (например, инструкции, расположенные внутри оператора цикла) сдвинуть еще на два пробела и т. д.

3.9. Общие сведения о вводе-выводе данных

Для обмена информацией с пользователем в Windows используются специальные формы, которые называются диалоговыми окнами.

В проектах VBA при организации диалога с пользователем используются две разновидности *встроенных диалоговых окон*: *окна сообщений* и *окна ввода*.

Окно ввода (InputBox) обеспечивает ввод информации, а *окно сообщений (MsgBox)* выводит сообщения для пользователя.

Кроме того, в программах VBA, написанных для приложения MS Excel, есть возможность ввод и вывод данных осуществить непосредственно из ячеек (в ячейки) рабочего листа.

3.10. Использование окна ввода

Ввод – это процесс переноса информации с внешнего носителя (клавиатуры, файла) в оперативную память компьютера.

Программная поддержка окна ввода обеспечивается функцией *InputBox()*, которая выполняет следующее:

- выводит на экран диалоговое окно, содержащее сообщение, поле ввода и две кнопки: *OK* и *Cancel* (рис. 10);



Рис. 10. Пример окна ввода

- устанавливает режим ожидания ввода текста пользователем и нажатия кнопки;
- возвращает значение типа *String* при нажатии кнопки *OK*, содержащее текст, введенный в поле ввода;
- возвращает пустую строку (значение *Empty*) при нажатии кнопки *Cancel*.

Синтаксис функции ввода (параметры, указанные в квадратных скобках, могут быть опущены): *InputBox(Prompt [, Title] [, Default] [, Xpos] [, Ypos] [, Helpfile, Context])*.

При записи функции используются следующие параметры:

- *Prompt* – текст сообщения, т. е. строковое выражение, отображаемое как сообщение в диалоговом окне. Может содержать несколько строк. Максимальная длина строки – 1024 символа.

- *Title* – заголовок диалогового окна. Если этот параметр опущен, то в строку заголовка помещается имя приложения.

- *Default* – текст, появляющийся в поле ввода до действий пользователя.

- *Xpos* – расстояние между левым краем экрана и левой границей диалогового окна.

- *Ypos* – расстояние между верхним краем экрана и верхней границей диалогового окна.

- *Helpfile, Context* – файл-справка и номер раздела в нем.

Наличие запятых, соответствующих отсутствующим аргументам, является обязательным.

Например, с помощью инструкции

x = InputBox("Введи значение x", "Пример 3.5")

появится окно ввода (см. рис. 10), в котором осуществляется ввод некоторого значения с клавиатуры, а затем введенное значение преобразуется в соответствии с типом переменной *x*.

При вводе вещественных чисел используется знак десятичная запятая «,».

3.11. Использование окна сообщений

Вывод – это процесс переноса информации из оперативной памяти компьютера на внешний носитель (экран, принтер, файл).

Практически любое информационное сообщение или предупреждение пользователь может вывести в специальном диалоговом окне.

Программная поддержка окна сообщения осуществляется встроенной процедурой *MsgBox*, которая выполняет следующее:

- выводит на экран диалоговое окно, содержащее сообщение и одну или более кнопок (кнопка *ОК* выводится всегда, остальные кнопки – по усмотрению разработчика);
- устанавливает режим ожидания нажатия кнопки пользователем.

Синтаксис процедуры вывода (параметры, указанные в квадратных скобках, могут быть опущены):
MsgBox Prompt [, *Buttons*]
 [, *Title*] [, *Helpfile*, *Context*].

При записи функции используются следующие параметры:

- *Prompt* – текст сообщения, т. е. строковое выражение, отображаемое как сообщение в диалоговом окне. Максимальная длина строки – 1024 символа.
- *Buttons* – числовое значение, представляющее сумму констант, указанных в таблицах 4–6.
- *Title* – заголовок диалогового окна. Если этот параметр опущен, то в строку заголовка помещается имя приложения.
- *Helpfile*, *Context* – файл-справка и номер раздела в нем.

Таблица 4. Значения параметра *Buttons* процедуры *MsgBox*, определяющие отображаемые кнопки в диалоговом окне

Имя константы	Значение константы	Отображаемые кнопки
vbOKOnly	0	ОК
vbOKCancel	1	ОК, Отмена
vbAbortRetryIgnore	2	Стоп, Повтор, Пропустить
vbYesNoCancel	3	Да, Нет, Отмена
vbYesNo	4	Да, Нет
vbRetryCancel	5	Повтор, Отмена

Таблица 5. Значения параметра *Buttons* процедуры *MsgBox*, определяющие отображаемые информационные значки в диалоговом окне

Имя константы	Значение константы	Значок сообщения
vbCritical	16	
vbQuestion	32	
vbExclamation	48	
vbInformation	64	

Таблица 6. Значения параметра *Buttons* процедуры *MsgBox*, определяющие основную кнопку в диалоговом окне

Имя константы	Значение константы	Номер основной кнопки
vbDefaultButton1	0	1
vbDefaultButton2	256	2
vbDefaultButton3	512	3
vbDefaultButton4	768	4

Параметры *Buttons* и *Title* могут отсутствовать. Однако, если в процедуре используется только параметр *Title*, то перед ним необходимо указать два разделительных символа «»» (запятые).

Пример 1. Вывод простого информационного сообщения. Процедура Pr_1:

```
Sub Pr_1()  
    ' Простое информационное сообщение  
    MsgBox "Привет!!!"  
End Sub
```

Результатом работы данной процедуры является информационное сообщение, выведенное в виде диалогового окна (рис. 11а). Поскольку в процедуре *MsgBox* используется только параметр *Prompt*, то в диа-

логовом окне выведена только одна обязательная кнопка *ОК*, которая используется для закрытия окна, и в заголовке окна выведено имя приложения – Microsoft Excel.



Рис. 11. Четыре вида окна сообщений

Для вывода в строке заголовка окна сообщения личного заголовка (рис. 11б) необходимо обращение к процедуре *MsgBox* записать следующим образом:

MsgBox "Привет!!!", , "Пример"

Усложним выводимое информационное сообщение. На рис. 11в показан результат вывода строковой константы "Привет!!!" и значения строковой переменной *Name*. Для их вывода в качестве единого информационного сообщения использована операция конкатенации (сцепления). Данный вывод реализован с помощью следующей процедуры:

```
Sub Pr_1_б()
' Простое информационное сообщение
' с выводом заголовка
Dim Name As String
Name = "Анна"
MsgBox "Привет!!!" & Name, , "Пример"
End Sub
```

Для вывода сообщения в нескольких строках используется стандартная функция *Chr()*. Она позволяет получать символы, генерируемые при нажатии различных клавиш. Например, *Chr(9)* соответствует нажатию клавиши *Tab*, *Chr(13)* – клавиши *Enter*.

Поскольку символы, используемые для начала новой строки, являются очень важными при форматировании сообщений и других строковых данных, редактор VBA имеет несколько predefined констант для этих символов, чтобы не было необходимости использовать функцию *Chr()*:

- *vbCr* – символ возврата каретки, эквивалент выражения *Chr(13)*;
- *vbTab* – символ табуляции, эквивалент выражения *Chr(9)*.

Символы табуляции включают в строки для выравнивания данных в столбцах.

На рис. 11г приведены результаты работы процедуры *Pr_1_в()*:

```
Sub Pr_1_в()
' Простое информационное сообщение
' с выводом заголовка
Dim Name As String
Name = "Анна"
MsgBox "Привет!!! " & vbCr & Name, , "Пример"
End Sub
```

Пример 2. Вывод информационного сообщения совместно с информационным значком в окне сообщения.

Процедура *Pr_2* осуществляет вывод диалогового окна, представленного на рис. 12:

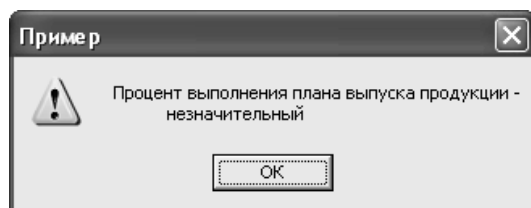


Рис. 12. Предупреждение

Данная процедура может быть записана следующим образом:

```
Sub Pr_2()  
    ' Вывод сообщения в сочетании с  
    ' предупреждающим информационным значком  
    MsgBox "Процент выполнения плана выпуска продукции -" & _  
        vbCr & " незначительный", vbExclamation, _  
        "Пример"  
End Sub
```



Для вывода информационного значка (Предупреждение) используется в качестве параметра *Атрибуты* встроенная константа *vbExclamation* (см. табл. 5).



Пример 3. Вывод диалогового окна с кнопками ОК (основная), Отмена и значком (Ответ с вопросом).

Окно сообщений, содержащее информационный значок "Вопрос", приведено на рис. 13.

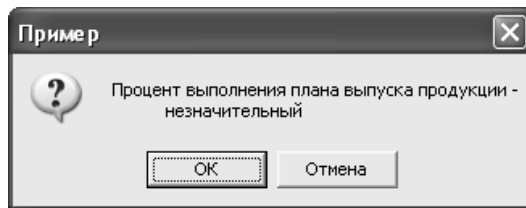


Рис. 13. Ответ с вопросом

Это окно можно вывести с помощью следующей процедуры:

```
Sub Pr2_3()  
    ' Вывод сообщения с двумя кнопками ОК и Отмена  
    ' в сочетании с  
    ' предупреждающим информационным значком.  
    ' Кнопка ОК является основной  
    MsgBox "Процент выполнения плана выпуска продукции -" & _  
        Chr(13) & " незначительный", vbOKCancel + _  
        vbQuestion + vbDefaultButton1, "Пример"  
End Sub
```

Пример 4. Реализация процедуры, осуществляющей диалог с Пользователем. На рис. 14 приведен текст процедуры "Privet".

```
Sub Privet()  
    Dim Name As String  
    Dim Reply As Long, Button As Long  
    Name = InputBox("Привет, как тебя зовут?", _  
        "Приветствие")  
    If Name = Empty Then  
        MsgBox "Жаль!", vbExclamation, "Приветствие"  
        Exit Sub  
    Else  
        Button = vbQuestion + vbYesNo  
        Reply = MsgBox("Очень приятно, " & Name & _  
            "! Ты хочешь учиться?", _  
            Button, "Приветствие")  
        If Reply = vbYes Then  
            MsgBox "Читай книги!", vbInformation, "Приветствие"  
        Else  
            MsgBox "Думай!", vbCritical, "Приветствие"  
        End If  
    End If  
End Sub
```

Рис. 14. Текст процедуры "Privet"

Результат выполнения процедуры приведен на рис. 15. Здесь стрелки, ведущие от командных кнопок, указывают на диалоговые окна, появляющиеся при щелчке по этим кнопкам.

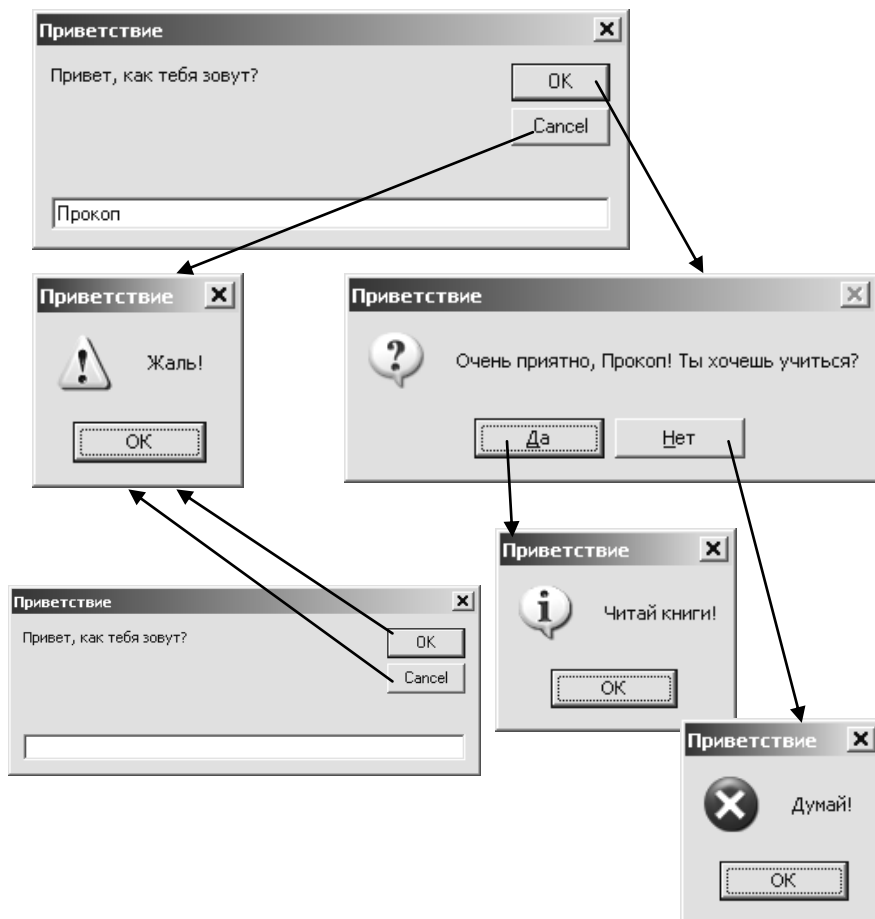


Рис. 15. Диалоговые окна в процессе выполнения

3.12. Обмен информацией с рабочим листом Excel

Для организации ввода (вывода) данных непосредственно с (на) рабочий лист Excel используется объект *Range* (Диапазон ячеек) или свойство *Cells* (Ячейка) объекта *Worksheet* (Рабочий лист). С их помощью можно работать с любой ячейкой рабочего листа Excel.

Используя объект *Cells*, можно содержимое ячейки рабочего листа присваивать в качестве значения переменной и наоборот, значение выражения выводить в любой ячейке:

$x = \text{Cells}(5,3).\text{Value}$

или

$x = \text{Cells}(5,3)$

или

$x = \text{Range}("C5").$

В приведенных операторах переменной x присваивается значение свойства *Value* ячейки *C5*.

С помощью оператора $x = \text{Cells}(1+i, 1+j)$ переменной x присваивается значение ячейки, номер строки и столбца которой определяется значениями выражения $1+i$ и $1+j$ соответственно.

С помощью оператора $\text{Cells}(8,4) = x+2*y$ в ячейку *D8* будет введено значение выражения $x+2y$.

При выполнении операторов

$p = \text{InputBox}(\text{"Введи номер строки"})$

$q = \text{InputBox}(\text{"Введи номер столбца"})$

$\text{Cells}(p,q) = a + \text{Range}("A1")$

в ячейку рабочего листа, расположенную в строке p и столбце q , выводится значение выражения $a + \text{Range}("A1")$, при этом номера строки и столбца вводятся с клавиатуры.

Пример 5. Вычисление значения функции $y = (x+3)^2 + \frac{2a-3b}{x^2-2.8}$. Ввод и вывод данных с помощью рабо-

чего листа Excel.

Фрагмент рабочего листа с исходными данными представлен на рис. 16а. Расчеты получены с помощью процедуры *Pr2_4()* и выведены на тот же рабочий лист (рис. 16б).

	A	B	C	D
1	Исходные данные:			
2		a=	13	
3		b=	57	
4		x=	108	
5				
6				
7				

а)

	A	B	C	D
1	Исходные данные:			
2		a=	13	
3		b=	57	
4		x=	108	
5				
6	Значение функции:			
7			12320,99	
8				
9				

б)

Рис. 16. Ввод и вывод информации с использованием рабочего листа

Данная процедура может быть записана следующим образом:

Sub Pr2_4 ()

Dim a As Byte, b As Byte, x As Integer, y As Single

a = Cells (2, 3): b = Cells (3, 3): x = Cells (4, 3)

*y = (x + 3) ^ 2 + (2 * a - 3 * b) / (x ^ 2 - 2.8)*

Cells(6, 1) = "Значение функции:"

Cells (7, 3) = y

End Sub

4. ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ ПРОСТЫХ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ В СРЕДЕ VBA

4.1. Операторы цикла

Наиболее употребляемыми в среде VBA являются два оператора цикла: оператор цикла с предусловием и арифметический оператор цикла.

Оператор цикла с предусловием имеет следующую структуру:

While условие

последовательность операторов

Wend

Этот оператор выполняет *последовательность операторов* до тех пор, пока *условие* имеет значение *True*.

Арифметический оператор цикла имеет следующую структуру:

For счетчик = начальное значение To конечное значение [Step шаг]

последовательность операторов

Next счетчик

Этот оператор повторяет *последовательность операторов* до тех пор, пока *счетчик* изменяется от *начального значения* до *конечного значения* с указанным *шагом*. Если шаг не указан, то он принимается равным единице.

4.2. Табулирование функции

4.2.1. Табулирование в последовательность окон сообщений

На рис. 17 приведен пример записи процедуры табулирования функции $y = 2 \cdot x$ в последовательность диалоговых окон.

На рис. 18 приведены диалоговые окна ввода и сообщений, выводимые при выполнении процедуры, реализующей табулирование для заданного диапазона значений аргумента x : от начального $x_n = 0,3$ до конечного $x_k = 0,9$ с шагом $\Delta x = 0,4$.

```
Option Explicit
Sub ТабулированиеMsgBox()
' Табулирование в сменяемые друг друга окна сообщений
Dim x As Single, xn As Single
Dim xk As Single, dx As Single, y As Single
xn = InputBox("Введите начальное значение x", _
    "Табулирование")
xk = InputBox("Введите конечное значение x", _
    "Табулирование")
dx = InputBox("Введите шаг изменения x", _
    "Табулирование")
x = xn
While x < (xk + dx / 1000)
    y = 2 * x
    MsgBox "x=" & x & "    y=" & y
    x = x + dx
Wend
End Sub
```

Рис. 17. Текст первой процедуры табулирования

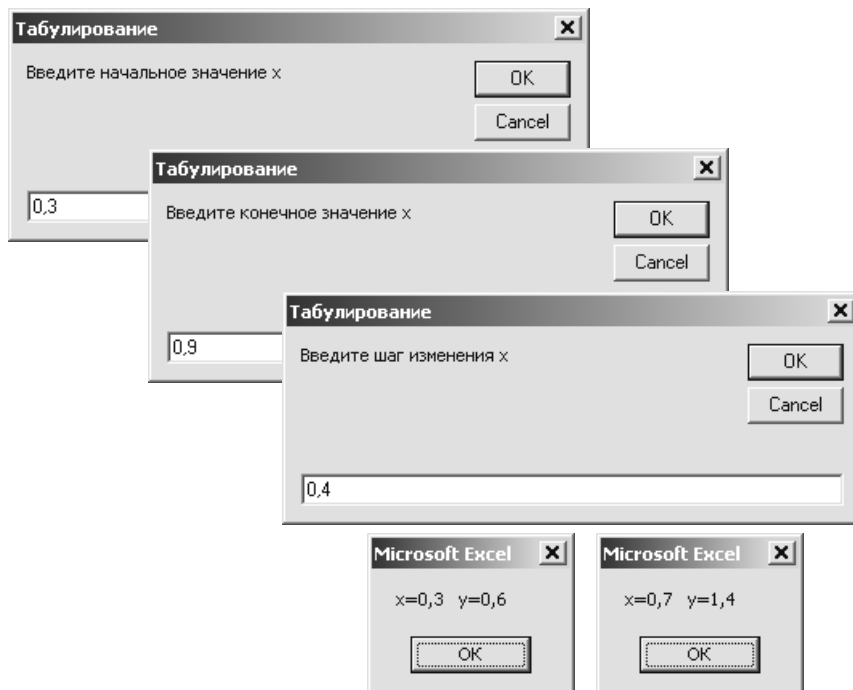


Рис. 18. Последовательность диалоговых окон процедуры табулирования функции

4.2.2. Табулирование в одно окно сообщений

На рис. 19 приведен пример записи процедуры, реализующей табулирование функции $y = 2 \cdot x$ для заданного диапазона значений аргумента x и выводящей результат в одно диалоговое окно. Для этого используется строковая переменная s , в которой с помощью операции конкатенации накапливаются строки результата. Переход к новой строке в окне сообщений реализуется с помощью константы $vbCr$. На этом же рисунке приведено итоговое окно сообщений.

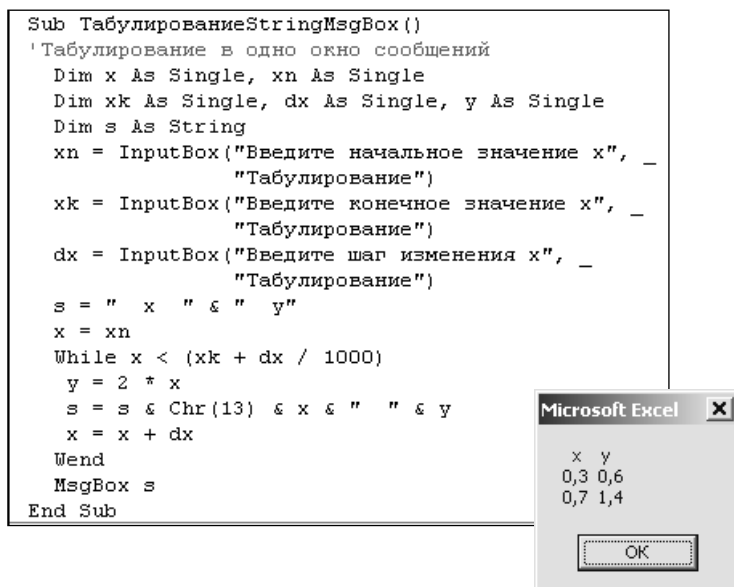


Рис. 19. Процедура табулирования функции в одно диалоговое окно

4.2.3. Табулирование в ячейки, задаваемые абсолютной адресацией

На рис. 20 приведен пример записи процедуры, реализующей табулирование функции $y = 2 \cdot x$ для заданного диапазона значений аргумента x и выводящей результат в ячейки рабочего листа Excel. Исходные данные вводятся с помощью трех окон ввода. После ввода вычисляется количество значений табуляции на

основе формулы $n = \left\lceil \frac{x_k - x_n}{\Delta x} \right\rceil + 1$. Здесь квадратные скобки означают округление до ближайшего меньшего целого. Адрес ячейки задается с помощью свойства *Cells(i, j)*, где *i* – номер строки, *j* – номер столбца.

```

Sub ТабулированиеЯчейки()
' Табулирование в ячейки, задаваемые абсолютной адресацией
Dim x As Single, xn As Single
Dim xk As Single, dx As Single, y As Single
Dim N As Integer, i As Integer
xn = InputBox("Введите начальное значение x", _
              "Табулирование")
xk = InputBox("Введите конечное значение x", _
              "Табулирование")
dx = InputBox("Введите шаг изменения x", _
              "Табулирование")

x = xn
N = Fix((xk - xn) / dx) + 1
Cells(1, 1) = "N"
Cells(1, 2) = "x"
Cells(1, 3) = "y"
For i = 1 To N Step 1
    y = 2 * x
    Cells(1 + i, 1) = i
    Cells(1 + i, 2) = x
    Cells(1 + i, 3) = y
    x = x + dx
Next i
End Sub

```

	A	B	C	D
1	N	x	y	
2		1	0,3	0,6
3		2	0,7	1,4
4				

Рис. 20. Процедура табулирования функции с выводом результатов в ячейки рабочего листа Excel

4.3. Описание числовых массивов

Массивы описываются с помощью оператора *Dim*.

Оператор *Dim A(3) As Single* объявляет массив *A*, состоящий из четырех переменных *A(0)*, *A(1)*, *A(2)*, *A(3)*.

Оператор *Dim B(1 To 12) As Integer* объявляет массив *B*, состоящий из 12 целых переменных.

Базовый индекс по умолчанию равен 0. Его можно изменить в конкретном описании массива, как это сделано для массива *B*, или с помощью инструкции *Option Base 1*, которая помещается в области описания модуля.

4.4. Ввод и вывод элементов массива

На рис. 21 приведен пример процедуры, организующей ввод и вывод в диалоговые окна элементов одномерного массива. Здесь стрелки, ведущие от операторов, указывают на диалоговые окна, появляющиеся при выполнении соответствующих операторов.

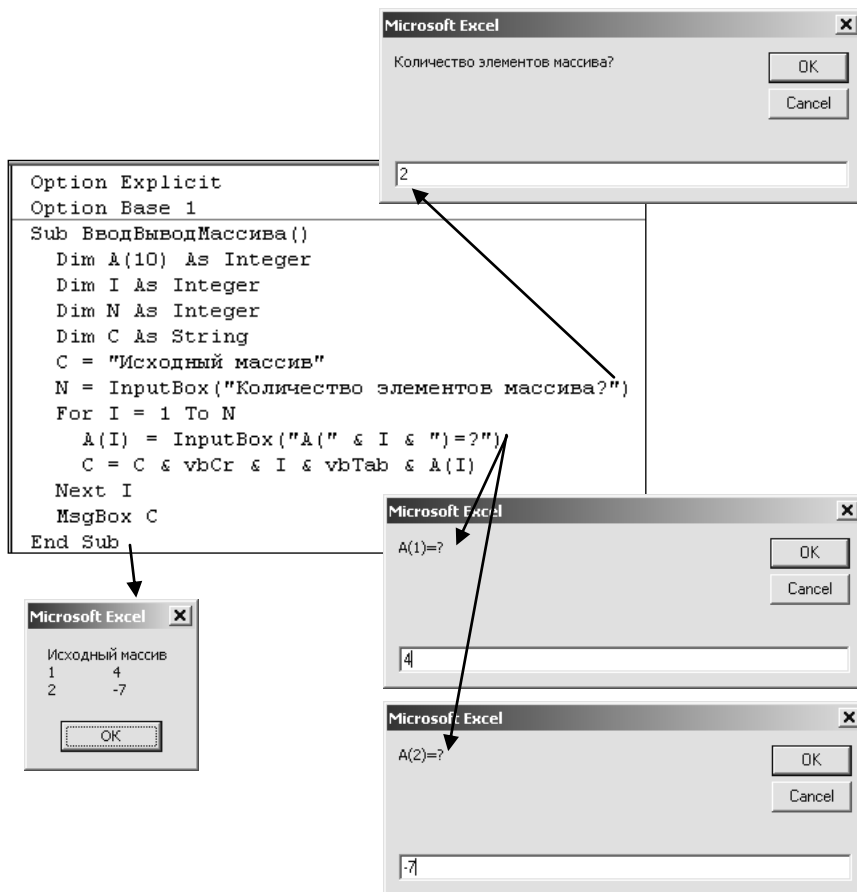


Рис. 21. Процедура ввода и вывода элементов одномерного массива

4.5. Простые операции над одномерными массивами

В качестве примера рассмотрим следующую задачу: для заданного целочисленного массива вычислить сумму квадратов элементов, больших заданного C , и количество неположительных чисел, стоящих на местах, кратных 4. Вывод результатов произвести в одно окно сообщений.

На рис. 22 приведен текст искомой процедуры.

```

Sub ПростыеОперацииМассив()
    Dim A(10) As Integer
    Dim I As Integer, N As Integer
    Dim C As Integer, Sum As Integer
    Dim K As Integer
    Dim S As String
    S = "Исходный массив" 'Строка для вывода результатов
    N = InputBox("Количество элементов массива?")
    For I = 1 To N
        A(I) = InputBox("A(" & I & ")=?")
        S = S & vbCrLf & I & vbTab & A(I)
    Next I
    C = InputBox("Введите число для сравнения")
    S = S & vbCrLf & "Число для сравнения" & vbCrLf & C
    Sum = 0 'Сумма квадратов элементов, больших C
    For I = 1 To N
        If A(I) > C Then
            Sum = Sum + A(I) ^ 2
        End If
    Next I
    S = S & vbCrLf & "Сумма квадратов элементов, больших " & C & " равна " & Sum
    K = 0 'Счетчик неположительных элементов,
    Rem стоящих на местах, кратных 4
    For I = 4 To N Step 4
        If A(I) <= 0 Then
            K = K + 1
        End If
    Next I
    S = S & vbCrLf & "Количество неположительных, " & K & " стоящих на местах, кратных 4 равно " & K
    MsgBox S
End Sub

```

Рис. 22. Процедура простой обработки одномерного массива

На рис. 23 приведены примеры диалоговых окон, которые появляются при запуске разработанной процедуры.

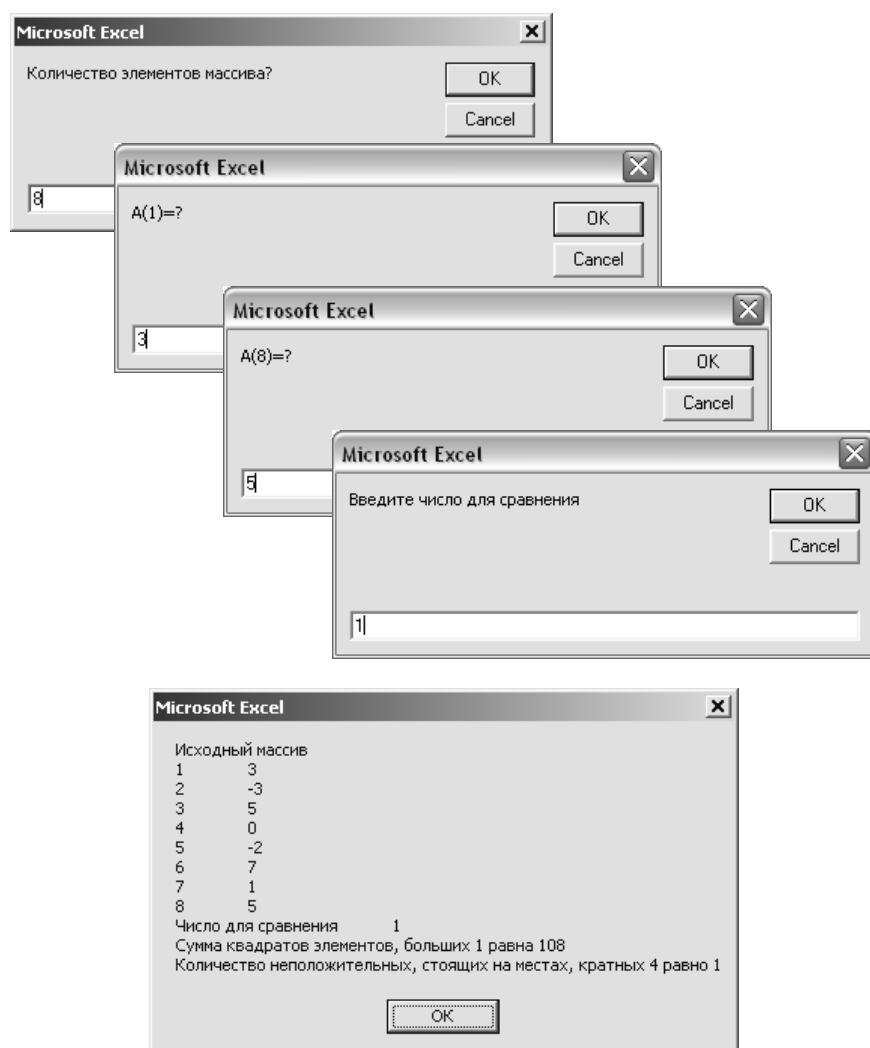


Рис. 23. Примеры диалоговых окон, появляющихся при запуске разработанной процедуры

4.6. Определение минимального и максимального элементов массива

В качестве примера рассмотрим задачу нахождения первого минимального элемента массива и перестановку его с предпоследним элементом массива. Текст искомой процедуры приведен на рис. 24.

```

Sub МинимальныйМассива()
    Dim A(10) As Integer
    Dim I As Integer, N As Integer
    Dim K As Integer, Min As Integer
    Dim S As String
    S = "Исходный массив" & vbCrLf 'Строка для вывода результатов
    N = InputBox("Количество элементов массива?")
    For I = 1 To N
        A(I) = InputBox("A(" & I & ")=?")
        S = S & A(I) & vbCrLf
    Next I
    Min = A(1) 'Первый элемент массива принимается за минимальный
    K = 1 'Номер первого минимального элемента
    For I = 2 To N
        If A(I) < Min Then
            Min = A(I)
            K = I
        End If
    Next I
    S = S & vbCrLf & "Минимальный элемент равен " & Min & _
        ", его номер равен " & K
    If N = 1 Then
        S = S & vbCrLf & "В массиве только один элемент"
    Else 'Перестановка элементов
        A(K) = A(N - 1)
        A(N - 1) = Min
        S = S & vbCrLf & "Полученный массив" & vbCrLf
        For I = 1 To N
            S = S & A(I) & vbCrLf
        Next I
    End If
    MsgBox S
End Sub

```

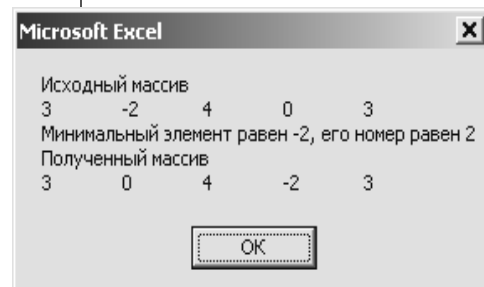


Рис. 24. Процедура нахождения минимального элемента массива и пример ее выполнения

4.7. Формирование массива

В качестве примера рассмотрим задачу формирования из заданного вещественного массива нового массива, в который следует включить только положительные элементы исходного массива. Текст искомой процедуры приведен на рис. 25.

```

Sub ФормированиеМассива()
    Dim A(10) As Single, B(10) As Single
    Dim I As Integer, N As Integer
    Dim K As Integer
    Dim S As String
    S = "Исходный массив" & vbCrLf 'Строка для вывода результатов
    N = InputBox("Количество элементов массива?")
    For I = 1 To N
        A(I) = InputBox("A(" & I & ")=?")
        S = S & A(I) & vbCrLf
    Next I
    K = 0 'Количество элементов в новом массиве
    For I = 1 To N
        If A(I) > 0 Then
            K = K + 1
            B(K) = A(I)
        End If
    Next I
    S = S & vbCrLf & "Количество элементов в новом массиве равно " & K
    If K = 0 Then
        S = S & vbCrLf & "В исходном массиве нет положительных элементов"
    Else
        S = S & vbCrLf & "Полученный массив" & vbCrLf
        For I = 1 To K
            S = S & B(I) & vbCrLf
        Next I
    End If
    MsgBox S
End Sub

```

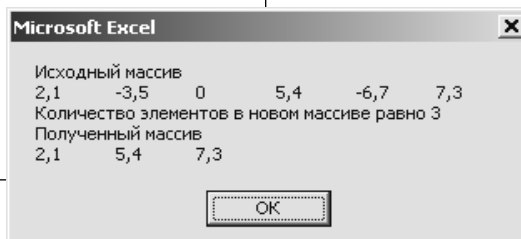


Рис. 25. Процедура формирования массива и пример ее выполнения

5. ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ ВЛОЖЕННЫХ ЦИКЛОВ

5.1. Вложенные циклы

Циклы разрешается помещать внутри других циклов, подобно тому, как условные операторы вкладываются друг в друга.

Циклы называются *вложенными*, если один из них целиком входит в рабочую часть другого, причем циклы могут быть разными: *For* или *While*.

При вложении циклов необходимо, чтобы каждый цикл имел свою уникальную переменную – параметр цикла (счетчик).

Пример 1. Таблица умножения чисел от 1 до 9 вида (рис. 26) реализуется посредством процедуры, текст которой приводится на рис. 27. Данный пример является частным случаем табулирования функции двух переменных $Z=f(i,j)$, где каждая из переменных i и j изменяется от 1 до 9 с шагом 1.

	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1	1	2	3	4	5	6	7	8	9
3	2	2	4	6	8	10	12	14	16	18
4	3	3	6	9	12	15	18	21	24	27
5	4	4	8	12	16	20	24	28	32	36
6	5	5	10	15	20	25	30	35	40	45
7	6	6	12	18	24	30	36	42	48	54
8	7	7	14	21	28	35	42	49	56	63
9	8	8	16	24	32	40	48	56	64	72
10	9	9	18	27	36	45	54	63	72	81

Рис. 26. Таблица умножения чисел от 1 до 9 на листе Excel

```

Sub Umn()
    Таблица умножения
    Dim i As Integer, j As Integer
    For i = 1 To 9
        Cells(i + 1, 1) = i
        For j = 1 To 9
            Cells(1, j + 1) = j
            Cells(i + 1, j + 1) = i * j
        Next j
    Next i
End Sub

```

Рис. 27. Процедура, реализующая таблицу умножения чисел от 1 до 9

5.2. Двухмерные массивы и их описание

Двухмерный массив (матрица) – это массив, элементы которого имеют два измерения: первый индекс – номер строки, а второй – номер столбца. Рис. 28 иллюстрирует принцип индексации двухмерного массива на примере массива размером 3×3 .

Номера строк	Номера столбцов		
	1	2	3
1	X_{11}	X_{12}	X_{13}
2	X_{21}	X_{22}	X_{23}
3	X_{31}	X_{32}	X_{33}

Рис. 28. Общий вид двухмерного массива

Если число строк равно числу столбцов, то матрица называется *квадратной*. У такой матрицы, имеющей размер $n \times n$, можно выделить главную диагональ, на которой лежат элементы X_{ij} , обладающие свойством $i=j$, и побочную диагональ, элементы которой обладают свойством $i+j=n+1$. Для остальных элементов выполняются следующие условия:

- при $i < j$ элемент находится выше главной диагонали;
- при $i > j$ – ниже главной диагонали;
- при $i + j < n + 1$ – выше побочной диагонали;
- при $i + j > n + 1$ – ниже побочной диагонали.

Объявление (описание) матриц аналогично объявлению одномерных массивов: в скобках через запятую указывается число строк и число столбцов или пределы изменения индексов.

Пример 2. Объявление матрицы размерностью 10×8, состоящей из вещественных чисел может быть записано следующим образом:

Option Base 1
Dim Z (10,8) As Single

Пример 3. Объявление матрицы, размерностью 5×4, состоящей из целых чисел, записывается следующим образом:

Dim B (1 To 5, 1 To 4) As Integer

5.3. Ввод и вывод двумерных массивов

Ввод матрицы построчно может быть выполнен с помощью процедуры, фрагмент которой приведен на рис. 29.

```
n = Inputbox("Введи число строк", "Ввод размерности массива")
m = Inputbox("Введи число столбцов", "Ввод размерности массива")
For i=1 To n
  For j=1 to m
    X(i,j)=Inputbox("Введи X(" & i & ", " & j & ")", "Ввод элемента матрицы")
  Next j
Next i
```

Рис. 29. Программный код для построчного ввода матрицы

Вывод матрицы построчно на лист рабочей книги Excel может быть выполнен с помощью процедуры, фрагмент которой приведен на рис. 30.

```
Cells(1,1)="Матрица X"
For i=1 To n
  For j=1 to m
    Cells(i+1,j)=X(i,j)
  Next j
Next i
```

Рис. 30. Программный код для построчного вывода матрицы

На рис. 31 представлен результат работы этого фрагмента для ранее введенной матрицы.

	А	В	С
1	Матрица X		
2	2	4	5
3	0	3	77
4	5	2	1
5	0	8	-5

Рис. 31. Пример построчного вывода матрицы

5.4. Алгоритмы обработки двумерных массивов

Основные принципы записи алгоритмов обработки двумерных массивов рассмотрим на примере решения трех задач.

Задача 1. Определите сумму прибыли, полученную магазином в каждом месяце, по заданным значениям прибыли для отдельных видов товара. Будем считать, что исходные данные заданы электронной таблицей, приведенной на рис. 32.

	А	В	С	Д	Л	М
1	Прибыль от продажи товаров					
2	Товар	Прибыль, тыс. р.				
3		январь	февраль	март	...	декабрь
4	Сахар	1550	1430	1720		1540
5	Мука	2300	2200	2610		3140
6	Молоко	1860	1570	2020		2190
7	Колбаса	2400	2130	2590		2980
8	Итого	8110	7330	8940	...	9850

Рис. 32. Электронная таблица как пример двумерного массива

Решение

Обозначим P_{ij} – прибыль, полученную от продажи i -го товара в j -м месяце ($i=1, 2, \dots, 12$; $j=1, 2, 3, 4$), $PM(j)$ – прибыль в j -м месяце. Искомая процедура приведена на рис. 33.

```
Option Explicit
Option Base 1
Sub PribMes()
    Dim P(4, 12) As Single
    Dim PM(12) As Single
    Dim I As Integer, J As Integer
    Rem ввод в массив P данных из ячеек листа
    For I = 1 To 4
        For J = 1 To 12
            P(I, J) = Cells(I + 3, J + 1)
        Next J
    Next I
    Rem Вычисление суммы прибыли для каждого товара
    For I = 1 To 4
        PM(I) = 0
        For J = 1 To 12
            PM(I) = PM(I) + P(I, J)
        Next J
    Next I
    Rem Вывод результатов в ячейки листа
    For I = 1 To 4
        Cells(I + 3, 14) = PM(I)
    Next I
End Sub
```

Рис. 33. Процедура решения задачи 1

Задача 2. Для данных, приведенных на рис. 32, определите номера месяца и товара, при которых магазин получил наибольшую прибыль.

Решение

Введем следующие обозначения для переменных процедуры:

- max – максимальная полученная прибыль;
- n_max – номер месяца, в котором она получена;
- n_tow – номер товара, для которого получена максимальная прибыль.

Фрагмент процедуры для решения задачи приводится на рис. 34.

```
max = P(1, 1)
n_max = 1
n_tow = 1
For i = 1 To 4
    For j = 1 To 4
        If P(i, j) > max Then
            max = P(i, j)
            n_max = i
            n_tow = j
        End If
    Next j
Next i
```

Рис. 34. Программный код для задачи 2

Задача 3. Для заданной целочисленной квадратной матрицы В размером $N \times N$ определите, в какой части матрицы больше положительных чисел: выше главной диагонали или ниже ее.

Решение

Обозначим $K1$ и $K2$ количество положительных чисел, лежащих выше и ниже главной диагонали.

Будем считать, что заданная матрица размером 4×4 располагается начиная с ячейки $A12$. Программный код для решения данного варианта представлен на рис. 35.

```

Sub PositMatr()
    Dim B(5, 5) As Integer
    Dim I As Integer, J As Integer
    Dim N As Integer, K1 As Integer, K2 As Integer
    N = 4
    For I = 1 To N
        For J = 1 To N
            B(I, J) = Cells(I + 11, J)
        Next J
    Next I
    K1 = 0: K2 = 0
    For I = 1 To N
        For J = 1 To N
            If (I < J) And (B(I, J) > 0) Then K1 = K1 + 1
            If (I > J) And (B(I, J) > 0) Then K2 = K2 + 1
        Next J
    Next I
    If K1 > K2 Then
        MsgBox "Положительных чисел над главной диагональю больше"
    ElseIf K1 < K2 Then
        MsgBox "Положительных чисел под главной диагональю больше"
    Else
        MsgBox "Одинаковое количество положительных" _
            & "над и под главной диагональю"
    End If
End Sub

```

Рис. 35. Программный код для задачи 3

6. ТЕХНОЛОГИИ ОБРАБОТКИ СИМВОЛЬНЫХ ДАННЫХ

Рассмотрим встроенные функции языка VBA, предназначенные для обработки строк.

Первые две функции предназначены для преобразования регистра всех символов, находящихся в строке-аргументе:

- 1) *LCase(string)* преобразует символы строки в символы нижнего регистра;
- 2) *UCase(string)* преобразует символы строки в символы верхнего регистра.

На рис. 36 приведена процедура, использующая указанные функции, и окно сообщений с примером выполнения процедуры.

Следующие четыре функции предназначены для выделения подстроки из строки-аргумента:

- 1) *Len(string)* возвращает количество символов в строке;
- 2) *Left(string, n)* выделяет подстроку, состоящую из *n* первых символов исходной строки;

```

Sub СтроковыеФункции1()
    Dim s As String, s1 As String, s2 As String
    s = InputBox("Введите строку")
    s1 = LCase(s)
    s2 = UCase(s)
    MsgBox "s=" & s & vbCrLf & "s1=" & s1 & vbCrLf & "s2=" & s2
End Sub

```

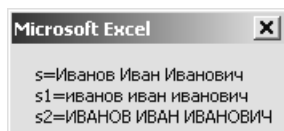


Рис. 36. Пример использования функций преобразования регистра

- 3) *Right(string, n)* выделяет подстроку, состоящую из *n* последних символов исходной строки;
- 4) *Mid(string, m, n)* выделяет подстроку, состоящую из *n* символов, начиная с *m*-го символа исходной строки.

На рис. 37 приведен фрагмент процедуры, использующей четыре указанные функции для выделения фамилии, имени и отчества из строки «Иванов Иван Иванович».

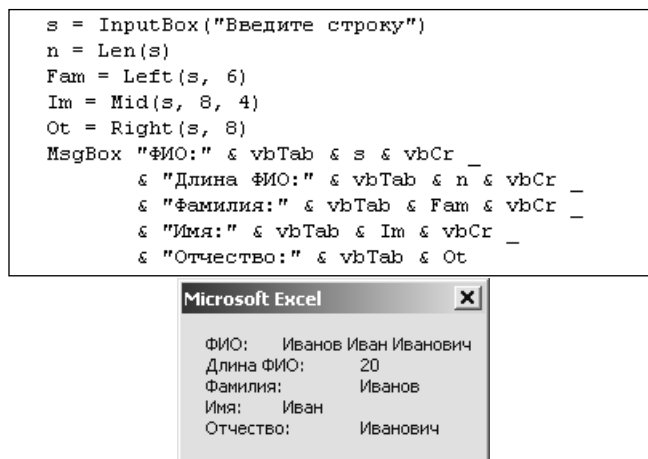


Рис. 37. Пример использования функций выделения подстроки

Далее рассмотрим три функции удаления пробелов из строки-аргумента:

- 1) *Ltrim(string)* возвращает копию строки без пробелов в ее начале;
- 2) *Rtrim(string)* возвращает копию строки без пробелов в ее конце;
- 3) *Trim(string)* возвращает копию строки без пробелов в ее начале и конце.

На рис. 38 приведено окно ввода для процедуры (рис. 39), использующей рассмотренные функции. На рис. 40 показано окно сообщений с результатом выполнения.

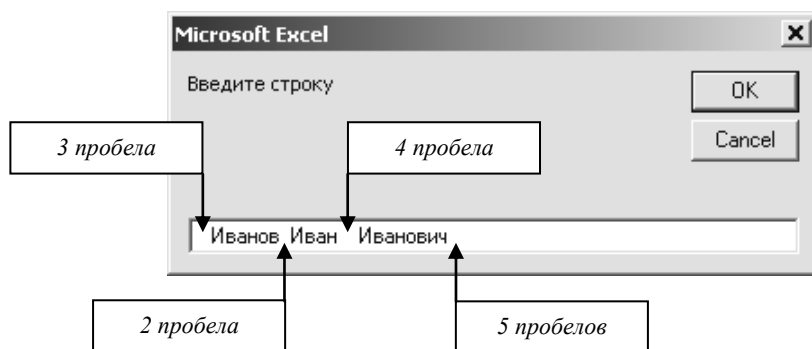


Рис. 38. Окно ввода для процедуры удаления пробелов

В языке VBA имеются две функции поиска подстроки в строке:

1) *InStr(m, str, pstr, k)* возвращает позицию первого вхождения подстроки *pstr* внутри строки *str*. Поиск начинается с позиции *m*. Параметр *k* (способ сравнения строк) может принять значение 0 (для двоичного сравнения) или 1 (для посимвольного сравнения без учета регистра). Параметры *m* и *k* могут быть опущены. Если параметр *m* опущен, то поиск начинается с первой позиции. По умолчанию параметр *k* равен 0;

2) *InStrRev(str, pstr, m, k)* возвращает позицию последнего вхождения подстроки *pstr* внутри строки *str*. Поиск начинается с позиции *m* и ведется справа налево. Параметр *k* (способ сравнения строк) может принять значение 0 (для двоичного сравнения) или 1 (для посимвольного сравнения без учета регистра). Параметры *m* и *k* могут быть опущены. Если параметр *m* опущен, то поиск начинается с последней позиции. По умолчанию параметр *k* равен 0.

Для иллюстрации использования последних двух функций приведем процедуру (рис. 41), выделяющую из заданной строки фамилию, имя и отчество.

```

Sub СтрочковыеФункции3()
    Dim s As String, s1 As String, s2 As String
    Dim s3 As String, n As Integer
    Dim n1 As Integer, n2 As Integer, n3 As Integer
    s = InputBox("Введите строку")
    n = Len(s)
    s1 = LTrim(s)
    n1 = Len(s1)
    s2 = RTrim(s)
    n2 = Len(s2)
    s3 = Trim(s)
    n3 = Len(s3)
    MsgBox "s=" & vbCrLf & "#" & s & "#" & vbCrLf _
        & "Исходная длина " & vbCrLf & n & vbCrLf _
        & "Без левых пробелов:" & vbCrLf _
        & "#" & s1 & "#" & vbCrLf _
        & "длина " & vbCrLf & n1 & vbCrLf _
        & "Без правых пробелов:" & vbCrLf _
        & "#" & s2 & "#" & vbCrLf _
        & "длина " & vbCrLf & n2 & vbCrLf _
        & "Без левых и правых пробелов:" _
        & vbCrLf & "#" & s3 & "#" & vbCrLf _
        & "длина " & vbCrLf & n3
End Sub

```

Рис. 39. Текст процедуры удаления пробелов

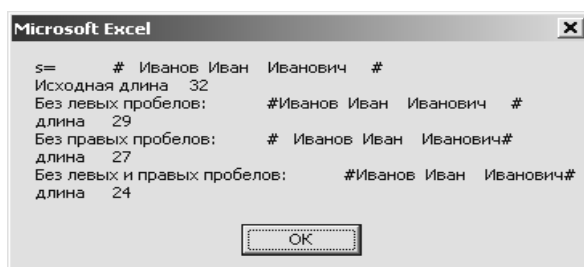


Рис. 40. Результат выполнения процедуры удаления пробелов

```

Sub СтрочковыеФункции4()
    Dim s As String, s1 As String, Fam As String
    Dim Im As String, Ot As String
    Dim n As Integer, n1 As Integer
    Dim n2 As Integer, n3 As Integer
    s = InputBox("Введите строку")
    n = Len(s)
    s1 = Trim(s)
    n1 = Len(s1)
    n2 = InStr(1, s1, " ", 1)
    n3 = InStrRev(s1, "|", , 1)
    Fam = Left(s1, n2 - 1)
    Ot = Right(s1, n1 - n3)
    Im = Mid(s1, n2 + 1, n3 - n2 - 1)
    Im = Trim(Im)
    MsgBox "s=" & vbCrLf & "#" & s & "#" & vbCrLf _
        & "Исходная длина " & vbCrLf & n & vbCrLf _
        & "Без левых и правых пробелов:" & vbCrLf _
        & "#" & s1 & "#" & vbCrLf _
        & "длина " & vbCrLf & n1 & vbCrLf _
        & "Фамилия:" & vbCrLf & "#" & Fam & "#" & vbCrLf _
        & "Имя:" & vbCrLf & "#" & Im & "#" & vbCrLf _
        & "Отчество:" & vbCrLf & "#" & Ot & "#"
End Sub

```

Рис. 41. Процедура выделения компонентов имени

Составим процедуру подсчета в заданной строке количества символов «+». На рис. 42 приведены два варианта процедуры решения этой задачи: первая – с помощью функции выделения подстроки *Mid*, а вторая – на основе функции *InStr*.

На рис. 43 приведены два окна сообщений, иллюстрирующих результаты выполнения процедур подсчета символов «+».

На рис. 44 приведен еще один вариант процедуры подсчета в заданной строке количества символов «+», на этот раз с использованием функции *InStrRev*.


```

Sub Строчковыефункции5()
    Dim s As String
    Dim n As Integer, i As Integer, k As Integer
    s = InputBox("Введите строку")
    n = Len(s)
    k = 0
    For i = 1 To n
        If Mid(s, i, 1) = "+" Then
            k = k + 1
        End If
    Next i
    MsgBox "s=" & vbCrLf & "#" & s & "#" & vbCrLf _
        & "Исходная длина " & vbCrLf & n & vbCrLf _
        & "Количество символов +:" & vbCrLf & k
End Sub

Sub Строчковыефункции6()
    Dim s As String, n As Integer
    Dim i As Integer, k As Integer, m As Integer
    s = InputBox("Введите строку")
    n = Len(s)
    k = 0
    i = 1
    While i <= n
        m = InStr(i, s, "+", 1)
        If m <> 0 Then
            k = k + 1
            i = m + 1
        Else
            i = n + 1
        End If
    Wend
    MsgBox "s=" & vbCrLf & "#" & s & "#" & vbCrLf _
        & "Исходная длина " & vbCrLf & n & vbCrLf _
        & "Количество символов +:" & vbCrLf & k
End Sub

```

Рис. 42. Процедуры подсчета символов «+» на основе функций *Mid* и *InStr*



Рис. 43. Результаты подсчета символов «+» в заданной строке

```

Sub Строчковыефункции7()
    Dim s As String, n As Integer
    Dim i As Integer, k As Integer, m As Integer
    s = InputBox("Введите строку")
    n = Len(s)
    k = 0
    i = n
    While i >= 1
        m = InStrRev(s, "+", i, 1)
        If m <> 0 Then
            k = k + 1
            i = m - 1
        Else
            i = 0
        End If
    Wend
    MsgBox "s=" & vbCrLf & "#" & s & "#" & vbCrLf _
        & "Исходная длина " & vbCrLf & n & vbCrLf _
        & "Количество символов +:" & vbCrLf & k
End Sub

```

Рис. 44. Процедура подсчета символов «+» на основе функции *InStrRev*

Функция нахождения подстроки в заданной строке *Replace(str, pstr, nstr, m, n, k)* находит и заменяет в строке *str* подстроку *pstr* подстрокой *nstr*. Поиск начинается с позиции *m*. Если параметр *m* опущен, то поиск начинается с первой позиции. Параметр *n* указывает, сколько подстрок надо заменить. Если этот параметр опущен, то будут произведены все замены. Параметр *k* (способ сравнения строк) может принять значение 0 (для двоичного сравнения) или 1 (для посимвольного сравнения без учета регистра). Параметры *m* и *k* могут быть опущены. При *m* > 1 из исходной строки удаляются первые *m*-1 символы.

Для иллюстрации технологии использования данной функции составим процедуру (рис. 45) замены в заданной строке всех вхождений подстроки «++» на символ «*». На этом же рисунке приведено окно сообщений как пример выполнения искомой процедуры.

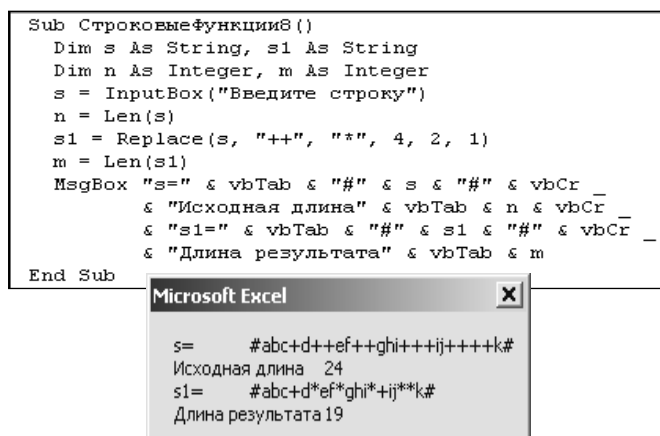


Рис. 45. Использование функции *Replace* для замены всех символов «++» на символ «*»

7. ТЕХНОЛОГИИ РАБОТЫ С ОБЪЕКТАМИ КЛАССА «ФОРМА»

7.1. Объект «Форма»: свойства и методы

С точки зрения удобства и частоты применения формы являются наиболее важными объектами, используемыми в приложениях MS Office. Они позволяют пользователю наиболее удобным образом общаться с программой, обеспечивая гибкую систему ввода и вывода данных, значительно сокращающую количество ошибок. Формы позволяют не только вводить новые значения исходных данных, но и просматривать и редактировать ранее введенные.

Редактор VBA позволяет создавать и использовать пользовательские окна при помощи объекта *UserForm*.

Объект *UserForm* – это пустое диалоговое окно. Его настройку можно выполнить путем добавления к объекту *UserForm* (пустое окно формы) элементов управления. Каждому объекту *UserForm* присущи определенные свойства, методы и события, которые он наследует от класса объектов *UserForm*.

Чтобы создать новое диалоговое окно, нужно сначала добавить к проекту новую форму (объект *UserForm*), а затем поместить в ее рабочую область элементы, необходимые для диалога программы с пользователем. Поскольку объекты *UserForm* хранятся в коллекции *UserForm* проекта, они являются частью проекта.

Для добавления к проекту новой формы нужно в редакторе VBA выполнить команду *Insert\UserForm*.

По умолчанию форма, добавляемая к проекту, получает имя *UserFormN* (*N*=1,2,...). Новая форма выводится в режиме разработки, что позволяет добавлять (или удалять) элементы управления, устанавливать свойства, методы формы и ее элементов управления и выполнять другие манипуляции с внешним видом формы в интерактивном режиме. Когда же форма введена и используется как часть выполняющейся программы, она находится в режиме выполнения. Форму можно переименовать, для чего в окне свойств (*Properties Windows*) нужно изменить свойство этого объекта. В режиме разработки вокруг формы выводится широкая рамка, а внутри – сетка из точек. Сетка помогает выравнивать и контролировать размеры элементов управления, которые можно выбирать и добавлять к форме с помощью панели инструментов *Toolbox*, изображенной на рис. 46. Панель, как правило, появляется только тогда, когда выбрана форма или один из ее элементов управления.



Рис. 46. Панель элементов управления объекта *Форма*

Работу с объектами класса *Форма* можно проверить в режиме выполнения с помощью команды *Run\Run* (или нажатием клавиши *F5*). При этом инициализируется только код, находящийся в модуле класса *Форма*).

Свойства формы можно устанавливать программным образом или через окно *Properties Windows*. К основным свойствам относятся следующие:

- *Name* – имя формы.
- *BackColor* – цвет заливки формы.
- *Caption* – заголовок формы.
- *ForeColor* – цвет переднего плана (цвета текста).
- *Cicle* (определяет, можно ли использовать клавишу *TAB* для перехода к следующему элементу формы).

К основным методам формы относятся следующие:

- *Copy* копирует выделенный в элементе управления текст в буфер обмена.
- *Cut* вырезает выделенный в элементе управления текст в буфер обмена.
- *Paste* вставляет содержимое буфера обмена в текущий элемент управления.
- *Show* выводит форму на экран, а если она не загружена в память, то сначала ее загружает и затем передает ей управление. Форма будет оставаться на экране до тех пор, пока не будет выполнен метод *Hide* этого объекта или форма не будет выгружена при помощи оператора *Unload*. Когда редактор VBA выполняет метод *Show*, то процедура, содержащая обращение к этому методу, приостанавливается до тех пор, пока выведенная форма не будет закрыта пользователем. При этом будут выполняться любые процедуры обработки событий в модуле класса *Форма*.










Загрузка формы осуществляется при помощи оператора *Load <имя формы>*, а выгрузка при помощи оператора *Unload <имя формы>*.

Оператор *Load* загружает форму в память и запускает метод формы *Initialize*, но не выводит форму на экран. Когда форма загружена, для работы с ней можно использовать написанную программу.

7.2. Элементы управления объекта «Форма»: их свойства и методы

Каждый элемент управления (ЭУ) – это объект с определенными свойствами, методами и событиями, которые пользователь может изменять либо программным путем, либо посредством окна *Properties Windows*. Список наиболее употребляемых стандартных кнопок приведен в табл. 7.

Таблица 7. Основные элементы управления объекта *Форма*

	Описание
	Надпись, метка (<i>Label</i>) служит для создания подписей к элементам управления
	Текстовое поле (<i>TextBox</i>) – окно для ввода текстовых данных (одно- или многострочное)
	Поле со списком (<i>ComboBox</i>) объединяет в себе окно редактирования текста и окно списка
	Список (<i>ListBox</i>) служит для выбора значений из списка
	Флажок (<i>CheckBox</i>) – элемент для включения (выключения) опции
	Кнопка-переключатель (<i>OptionButton</i>) выбирает одну из нескольких взаимоисключающих опций
	Выключатель (<i>ToggleButton</i>) работает так же как флажок, но переключается нажатием
	Рамка (<i>Frame</i>) объединяет (логически) элементы управления
	Командные кнопки (<i>CommandButton: CmdOK, CmdCancel...</i>) вызывают процедуры обработки

Обращение к элементам управления происходит в основном через их свойства, а также с помощью процедур обработки событий, написанных для каждого элемента. Некоторые из этих свойств приведены в табл. 8. Примеры описания событий для элементов управления описаны в подразделе 7.3 «Обработка события при работе с объектом *Форма*».

Таблица 8. Основные свойства элементов управления объекта *Форма*

Свойство	Где применяется	Описание
<i>Cancel</i>	<i>CommandButton</i>	Задаёт кнопку отмены диалогового окна. При нажатии на неё или кнопку <i>Esc</i> диалоговое окно исчезает
<i>BackColor</i>	Все ЭУ	Задаёт цвет фона ЭУ
<i>ForeColor</i>	Все ЭУ	Цвет переднего плана (текста)
<i>List</i>	<i>ComboBox</i>	Список, содержащийся в ЭУ
<i>Name</i>	Все ЭУ	Задаёт имя ЭУ
<i>Selected</i>	<i>ListBox</i>	Массив значений, выбранных из списка ЭУ
<i>Value</i>	Все ЭУ	Значение ЭУ
<i>Enabled</i>	Все ЭУ	Показывает, доступен (<i>True</i>) или нет (<i>False</i>) этот ЭУ
<i>Caption</i>	<i>Frame, Label, UserForm</i> и др.	Текст, отображаемый ЭУ, надпись на нём или рядом
<i>Default</i>	<i>CommandButton</i>	Определяет кнопку, заданную по умолчанию (она сработает, если нажать клавишу <i>Enter</i>)
<i>Font</i>	<i>Label, TextBox</i> и др.	Задаёт размер и тип шрифта
<i>TextAlign</i>	<i>Label, TextBox</i> и др.	Способ выравнивания текста (по левому краю, по центру или по правому краю)
<i>Width</i>	<i>Label, TextBox</i> и др.	Ширина окна
<i>Height</i>	<i>Label, TextBox</i> и др.	Высота окна

7.3. Обработка событий при работе с объектом «Форма»

Событие – это нечто такое, что может произойти с диалоговым окном или его элементом управления: щелчок на кнопке или переключателе, изменение содержимого окна редактирования или выбор элементов списка и т. п. Для организации правильной работы с диалоговыми окнами нужно предусмотреть реакцию программы на все возможные действия пользователя, если эти действия предусмотрены и не заблокированы. Поэтому создание диалоговых окон нужно начинать с разработки событийных процедур, т. е. процедур, реагирующих на эти события (процедур обработки событий). Их следует записывать в модуль класса, который является частью *UserForm*, в следующем виде:

<имя объекта>_<имя события>,

где <имя объекта> – имя формы или элементов управления;

<имя события> – имя события, которое нужно обработать.

В табл. 9 приведены основные события для объекта *Форма*, а в табл. 10 – для элементов управления.

Таблица 9. Наиболее часто используемые события объекта «Форма»

Событие	Синтаксис процедуры обработки	Описание
<i>Activate</i>	<i>Private Sub object_Activate ()</i>	Иницируется, когда окно формы становится активным. Это событие используется для обновления содержимого формы
<i>Click</i>	<i>Private Sub object_Click()</i>	Иницируется, когда по форме (любой ее части, не занятой ЭУ) щелкают мышью
<i>Initialize</i>	<i>Private Sub object_Initialize ()</i>	Иницируется, когда форма впервые загружается в память оператором <i>Load</i> или методом <i>Show</i> . Используется для инициализации ЭУ формы при ее появлении на экране
<i>Terminate</i>	<i>Private Sub object_Terminate()</i>	Иницируется, когда форма выгружается из памяти. Используется для осуществления служебных задач, которые необходимо выполнить прежде, чем переменные формы будут выгружены

Таблица 10. Наиболее часто используемые события элементов управления

Событие	Синтаксис процедуры обработки	Описание
<i>Change</i>	<i>Private Sub object_Change()</i>	Иницируется, когда изменяется значение ЭУ
<i>Click</i>	<i>Private Sub object_Click()</i>	Иницируется при щелчке на ЭУ и используется для выполнения действий при помощи кнопки

7.4. Использование форм для организации диалога с пользователем

Использование форм рассмотрим на примере решения приведенной ниже задачи. Необходимо рассчитать полную себестоимость продукции (S) по формуле

$$S = Z + R^{oo} + R^{мз} + R^н + R^в,$$

где Z – прямая заработная плата, р.;

R^{oo} – расходы на содержание и эксплуатацию оборудования, р.;

$R^{мз}$ – прямые материальные затраты, р.;

$R^н$ – накладные расходы, р.;

$R^в$ – внепроизводственные расходы, р.

Следует предусмотреть возможность вывода результатов в окно сообщений, на лист рабочей книги или одновременно – на лист и в окно.

Решение. Для ввода исходных данных воспользуемся формой, приведенной на рис. 47.

Рис. 47. Вид формы в режиме выполнения

В табл. 11 приводятся названия элементов формы и соответствующих им переменных в программном коде.

Решение задачи предполагает разработку следующих процедур:

- загрузки формы, которая создается в стандартном модуле, текст этой процедуры приведен на рис. 48;
- инициализации элементов управления формы, создаваемой в модуле *Ввод* класса *Формы* (рис. 49) и предназначенной для установки начальных значений элементов управления;
- обработки события (щелчка по форме), создаваемой в модуле *Ввод* класса *Формы* (рис. 50) и предназначенной для вывода сообщения о неточности действий пользователя;
- обработки щелчка по кнопке *Конец работы*, создаваемой в модуле *Ввод* класса *Формы* (рис. 50) и предназначенной для выгрузки формы;
- обработки щелчка по кнопке *Рассчитать*, создаваемой в модуле *Ввод* класса *Формы* (рис. 51) и предназначенной для выполнения требуемых расчетов.

Таблица 11. Наименования элементов и переменных

Тип элемента	Название в форме	Имя соответствующей переменной или свойства в программе
Форма	Ввод	Ввод
Метка	Поле1	–
Метка	Поле2	–
Метка	Поле3	–
Метка	Поле4	–
Метка	Поле5	–
Текстовое поле	x	ПЗ
Текстовое поле	y	R00

Тип элемента	Название в форме	Имя соответствующей переменной или свойства в программе
Текстовое поле	z	Rmз
Текстовое поле	a	Rн
Текстовое поле	b	Rв
Флажок	lst	lst
Флажок	win	win
Кнопка	Расчет_кнопка	Расчет_кнопка_Click()
Кнопка	Конец_кнопка	Конец_кнопка_Click()

```
Sub zap() ' Загрузка формы
    Load Ввод
    Ввод.Show
End Sub
```

Рис. 48. Процедура стандартного модуля загрузки формы

```
Private Sub UserForm_initialize()
' инициализация формы
' установка значений переменных по умолчанию
    x.Value = 0
    y.Value = 0
    z.Value = 0
    a.Value = 0
    b.Value = 0
    lst.Value = False
    win.Value = True
End Sub
```

Рис. 49. Процедура инициализации элементов управления формы

```
Private Sub UserForm_Click()
' Обработка события щелчок по форме
    MsgBox "Увы! Промазали..."
End Sub
Private Sub Конец_кнопка_Click()
```

```
Private Sub Расчет_кнопка_Click()
' Обработка события - щелчок по кнопке "Рассчитать"
    Dim ПЗ As Single, R00 As Single
    Dim Rмз As Single, Rн As Single, Rв As Single
    Dim S As Single
' Считывание значений переменных из формы
    ПЗ = x.Value
    R00 = y.Value
    Rмз = z.Value
    Rн = a.Value
    Rв = b.Value
' Вычисление себестоимости
    S = ПЗ + R00 + Rмз + Rн + Rв
' Проверка: выбран ли вывод в окно сообщений
    If win.Value Then
        MsgBox "Исходные данные, р.:" & vbCrLf & _
            & "Прямая заработная плата " & ПЗ & vbCrLf & _
            "Расходы на содержание и эксплуатацию оборудования " & _
            & R00 & vbCrLf & _
            "Прямые материальные затраты " & Rмз & vbCrLf & _
            & "Накладные расходы " & _
            Rн & vbCrLf & "Внепроизводственные расходы " & Rв & _
            & vbCrLf & "Результат:" & vbCrLf & _
            "Полная себестоимость продукции, р.: " & S & _
            , vbInformation, "Результаты расчета:"
    End If
' Проверка: выбран ли вывод на лист рабочей книги
    If lst.Value Then
        Range("A1:A8").Clear
        Cells(1, 1) = "Исходные данные, руб.:"
        Cells(2, 1) = "Прямая заработная плата " & ПЗ
        Cells(3, 1) = "Расходы на содержание и " & _
            & "эксплуатацию оборудования " & R00
        Cells(4, 1) = "Прямые материальные затраты " & Rмз
        Cells(5, 1) = "Накладные расходы " & Rн
        Cells(6, 1) = "Внепроизводственные расходы " & Rв
        Cells(7, 1) = "Результат:"
        Cells(8, 1) = "Полная себестоимость продукции, руб.: " & S
    End If
End Sub
```

Рис. 51. Процедура обработки щелчка по кнопке *Рассчитать*

ЗАДАНИЯ ЛАБОРАТОРНЫХ РАБОТ

Лабораторная работа 1. Создание и использование макросов

Цель работы: получение навыков создания макросов с абсолютными и относительными ссылками и их назначения различным объектам.

Задание 1. Создание и использование макросов с абсолютными ссылками

1. Вызовите MS Excel. Сохраните книгу в своей папке под именем *Макросы*.
2. Создайте таблицу по образцу табл. 12 и заполните в ней (без форматирования и обрамления) 7–8 строк данными. Расположите таблицу на листе MS Excel с ячейки *A1*.

Таблица 12. Образец таблицы для задания 1

Стоимость товаров

Курс доллара США – 2200 р.

№ п/п	Наименование товара	Цена, р.	Количество	Стоимость, р.	Стоимость, долл. США
1	Сметана	1200	26		
2	Молоко	650	15		
...					

В столбцы *№ п/п*, *Наименование товара*, *Цена*, *Количество* введите исходные данные, курс доллара введите в отдельную ячейку над таблицей, в первые ячейки столбцов *Стоимость, р.* и *Стоимость, долл. США* введите расчетные формулы и скопируйте их во все остальные ячейки соответствующих столбцов. Сохраните рабочую книгу на диске.

3. Переименуйте Лист1 в *Исходный*, Лист2 в *Рабочий*. Скопируйте таблицу на Лист2.

Работа с макросами выполняется только на Листе2 (Рабочий).

4. Запишите макрос для форматирования таблицы, для чего выполните команду *Сервис\Макрос\Начать запись*.

В результате появится диалоговое окно *Запись макроса*, в котором нужно выполнить следующие действия:

- указать имя макроса, например, *Форматирование*;
- ввести описание макроса (форматирование таблицы, дата, ФИО студента, группа);
- назначить сочетание клавиш для быстрого вызова макроса (использовать только латинские буквы);
- выбрать место для сохранения макроса *Эта книга*;
- нажать кнопку *ОК*.

Появится панель инструментов *Остановить запись*. Если панель инструментов не появилась, то вызовите ее на экран.

Выполните следующие команды по форматированию таблицы:

- для заголовка таблицы установите размер шрифта 14, полужирное начертание и выравнивание по центру таблицы;
- для значения курса доллара установите денежный формат;
- для содержимого таблицы установите размер шрифта 12;
- для заголовков граф таблицы установите начертание курсив и выравнивание по центру ячеек с переносом по словам;
- установите необходимую ширину столбцов таблицы;
- выполните обрамление таблицы.

После этого нажмите на кнопку *Остановить запись* или выполните команду *Сервис\Макрос\Остановить запись*.

5. Создайте макрос для удаления таблицы:

- выполните команду *Сервис\Макрос\Начать запись*;
- задайте необходимые параметры в диалоговом окне *Запись макроса* (имя макроса – *Удаление*), нажмите кнопку *ОК*;
- выделите таблицу;
- очистите содержимое и форматы ячеек с помощью команды *Правка\Очистить все*;
- восстановите стандартную ширину столбцов с помощью команды *Формат\Столбец\Ширина*, введите ширину 8,43, нажмите *ОК*;
- нажмите на кнопку *Остановить запись*;
- просмотрите текст созданных макросов в редакторе VBA, для чего выполните команду *Сервис\Макрос\Макросы*, выберите нужный макрос в списке, щелкните по кнопке *Изменить*;

- измените, при необходимости, текст макроса; сохраните изменения;
 - закройте окно редактора, а для этого выполните команду *File\Close and Return to Microsoft Excel* или нажмете комбинацию клавиш *ALT+Q* (или нажмете кнопку *Закрыть*);
 - вернитесь на рабочий лист Excel.
6. Выполните макросы и проверьте правильность их работы:
- очистите лист *Рабочий*;
 - скопируйте таблицу с листа *Исходный* на лист *Рабочий*;
 - выполните автоматическое форматирование таблицы: с помощью команды *Сервис\Макрос\Макросы* выберите в списке макрос *Форматирование*, щелкните по кнопке *Выполнить*;
 - удалите созданную таблицу с помощью макроса *Удаление*.
7. Создайте макрос *Итог* для добавления в конец таблицы двух итоговых строк: в одной должны подсчитываться суммы в двух последних графах таблицы, а в другой строке должно определяться минимальное значение в графе *Количество*. Проверьте работу макроса.
8. Сохраните книгу и завершите работу табличного процессора Excel.

Задание 2. Создание и использование макросов с относительными ссылками

1. Присвойте Листу3 книги *Макросы* имя *Геометрия*.
2. Создайте макрос для выполнения расчетов, указанных в табл. 13. Макрос должен обеспечивать выполнение следующих действий:
 - вывод заголовков над значениями исходных данных и результатов;
 - расчет результатов для введенных исходных данных;
 - вывод значений должен производиться начиная с любой активной ячейки.
3. Назначьте сочетание клавиш для быстрого вызова макроса.
4. Продемонстрируйте на листе *Геометрия* работу созданного макроса преподавателю.

Таблица 13. Варианты расчета параметров геометрических фигур и тел

Вариант	Вычисляемые параметры	Расчетные формулы
1	Площадь и периметр прямоугольника со сторонами a, b	$S = ab$ $P = 2(a+b)$
2	Полупериметр и площадь треугольника со сторонами a, b, c	$p = \frac{a+b+c}{2}$ $S = \sqrt{p \cdot (p-a) \cdot (p-b) \cdot (p-c)}$
3	Сторона и периметр квадрата со стороной a	$S = a^2$ $P = 4a$
4	Объем и площадь боковой поверхности параллелепипеда со сторонами a, b, c	$V = abc$ $S = 2(a+b)c$
5	Площадь кольца с внешним радиусом R и внутренним r	$S = \pi(R^2 - r^2)$
6	Объем и площадь поверхности куба со стороной a	$V = a^3$ $S = 6a^2$
7	Площадь боковой поверхности и объем цилиндра с радиусом основания r и высотой h	$S = 2\pi r h$ $V = \pi r^2 h$
8	Площадь и периметр прямоугольного треугольника с катетами a, b и гипотенузой c	$S = \frac{ab}{2}$ $P = a + b + c$
9	Высота и площадь параллелограмма со сторонами a, b и углом α	$h = a \cdot \sin \alpha$ $S = a \cdot b \cdot \sin \alpha$
10	Площадь основания и объем цилиндра с радиусом основания r и высотой h	$S = \pi r^2$ $V = Sh$
11	Объем и площадь основания параллелепипеда со сторонами a, b, c	$V = abc$ $S = ab$
12	Площадь основания и объем конуса с радиусом основания r и высотой h	$S = \pi r^2$ $V = \frac{Sh}{3}$
13	Площадь круга и длина окружности радиуса r	$S = \pi r^2$ $l = 2\pi r$
14	Площадь и периметр ромба со стороной a и углом α	$S = a^2 \cdot \sin \alpha$ $p = 4 \cdot a$
15	Гипотенуза и площадь прямоугольного треугольника с катетами a, b	$c = \sqrt{a^2 + b^2}$ $S = \frac{ab}{2}$
16	Высота и площадь равнобедренной трапеции с основаниями a, b ($b > a$) и углом при большем основании α	$h = \frac{b-a}{2} \operatorname{tg} \alpha$ $S = \frac{b+a}{2} h$

Задание 3. Назначение макроса графическому объекту

1. На листе *Геометрия* книги *Макросы* с помощью панели инструментов *Рисование* создайте графический объект, изображающий геометрическую фигуру, для расчета параметров которой создан макрос в соответствии с заданием 2.
2. Используйте для созданного графического объекта цветную заливку и тень.
3. Выберите команду *Назначить макрос* в контекстном меню графического объекта, выделите нужный макрос в списке макросов и щелкните по кнопке *ОК*.
4. Проверьте вызов макроса как с помощью щелчка по графическому объекту, так и клавишами быстрого вызова.

Задание 4. Назначение макроса стандартной кнопке

1. Активизируйте лист *Геометрия* книги *Макросы*.
2. Выполните команду *Вид\Панели инструментов*. Отметьте щелчком левой кнопки мыши панель *Формы*. На рабочем листе появится панель с элементами управления, выберите элемент *Кнопка* и на рабочем листе с помощью мыши определите его размер.
3. Как только будет отжата кнопка мыши, на экране появится диалоговое окно *Назначить макрос объекту*, в котором выберите имя макроса, созданного в задании 2.
4. В качестве имени кнопки используйте название фигуры или тела, параметры которого рассчитываются с помощью назначенного макроса.
5. Проверьте вызов макроса с помощью созданной стандартной кнопки, щелчка по графическому объекту и клавиш быстрого вызова.

Задание 5. Назначение макроса кнопке панели управления

1. Активизируйте лист *Геометрия* книги *Макросы*.
2. Выполните команду *Сервис\Настройка*. В диалоговом окне *Настройка* выберите вкладку *Команды*. В списке *Категории* выделите строку *Макросы*. В списке *Команды* щелкните на кнопке *Настраиваемая кнопка* и перетащите ее в область панелей инструментов.
3. При открытом окне *Настройка* с помощью команды *Изменить значок на кнопке* контекстного меню для новой кнопки создайте на ней рисунок, напоминающий геометрическую фигуру, для расчета параметров которой был создан макрос в задании 2.
4. При открытом окне *Настройка* с помощью команды *Назначить макрос* и поля *Имя* контекстного меню назначьте кнопке макрос и присвойте ей имя.
5. Проверьте вызов макроса с помощью созданной кнопки панели инструментов, стандартной кнопки, щелчка по графическому объекту и клавиш быстрого вызова.

Задание 6. Создание панели управления пользователя

1. Активизируйте лист *Геометрия* книги *Макросы*.
2. В дополнение к макросу, созданному в задании 2, создайте еще два макроса. В качестве заданий выберите такие два варианта из табл. 13, чтобы все три макроса выполняли расчеты для различных фигур и тел.
3. Создайте панель инструментов пользователя с именем *Геометрия*. Для этого выполните команду *Вид\Панели инструментов\Настройка*, вкладка *Панели инструментов*, команда *Создать*.
4. Поместите первую кнопку на созданной панели инструментов. Для этого выберите вкладку *Команды* в диалоговом окне *Настройка*, выделите в списке *Категории* строку *Макросы*, а в списке *Команды* – *Настраиваемая кнопка*. Перетащите изображение настраиваемой кнопки в область создаваемой панели инструментов пользователя. Смените стандартное изображение кнопки на изображение геометрической фигуры. Присвойте кнопке имя этой фигуры.
5. С помощью команды *Назначить макрос* выберите для кнопки соответствующий макрос.
6. Повторите пункты 4 и 5 еще для двух кнопок. Назначьте им соответствующие макросы и присвойте имена. При формировании изображений на кнопках используйте для каждой кнопки свой цвет.
7. Перетащите созданную панель с рабочего листа в область панелей инструментов.
8. Для присоединения панели инструментов к рабочей книге выполните команду *Вид\Панели инструментов\Настройка*, щелкните по кнопке *Вложить* на вкладке *Панели инструментов*, выделите панель пользователя *Геометрия* и щелкните по кнопке *Копировать*, а затем по кнопке *ОК*. Закройте окно *Настройка*. Сохраните файл.
9. Проверьте вызов макросов с помощью созданной панели инструментов.

Лабораторная работа 2. Создание математической пользовательской функции на основе линейного алгоритма

Цель работы: запись арифметических выражений на языке VBA.

В редакторе VBA создайте код функции в соответствии с заданным вариантом из табл. 14. В качестве аргументов функции должны выступать x , a , b . Произведите отладку созданной функции с помощью встроенных функций MS Excel. Пример решения задачи изложен в подразделах 2.1 «Правила набора пользовательских функций» и 2.2 «Математические пользовательские функции».

Таблица 14. Варианты математических функций

Вариант	Вид функции	Вариант	Вид функции
1	$y = \frac{\arctg bx}{1 + \sin^2 x}$	16	$y = \frac{\arctg (a + x)}{\sqrt{a^3 + b^3}}$
2	$y = \frac{\sin^2 x + a}{\sqrt{x + bx}}$	17	$y = \frac{1 + \sqrt{bx}}{0,5 + \sin^2 ax}$
3	$y = \sqrt{\frac{a + bx}{\ln^2 x}}$	18	$y = \frac{a - e^{bx}}{\ln 2x }$
4	$y = \frac{\ln^2(x - b)}{a\sqrt{x}}$	19	$y = \frac{(a + bx)^2}{1 + \cos^3 ax}$
5	$y = \frac{a \ln^2 x}{b + \sqrt{x}}$	20	$y = \frac{b + \sin^2 ax}{e^{-x/2}}$
6	$y = \frac{e^{ax} + b}{1 + \cos^2 x}$	21	$y = \frac{\sin^2 x - a}{bx}$
7	$y = \frac{a + \sqrt[3]{x}}{\sin^2 bx}$	22	$y = \frac{\arctg^2 ax}{b + 0,5x}$
8	$y = \frac{a\sqrt{ x } - bx}{\ln^3 x}$	23	$y = \frac{\ln(a^2 - x)}{b \sin^2 x}$
9	$y = \frac{\sqrt{ax - b}}{\lg^2 x}$	24	$y = \frac{a - \sqrt{bx}}{1 + \cos 2x }$
10	$y = e^{-x} \frac{a + bx}{\ln^2 x + 1 }$	25	$y = \frac{\ln^2(a + x)}{(b + x)^2}$
11	$y = \frac{\lg^2 x - b}{e^{ax}}$	26	$y = \frac{\sqrt{a \ln x}}{1 + \lg^2 bx}$
12	$y = \frac{\arctg bx}{1 + \sqrt[3]{ax}}$	27	$y = \frac{1 + \lg^2 x}{b + e^{x/a}}$
13	$y = \frac{\sin^3 ax}{ax + b}$	28	$y = \frac{\cos^2 2x + b}{\sqrt{1 + e^{ax}}}$
14	$y = \frac{e^{-ab}}{b + \cos^3 ax}$	29	$y = \frac{\sqrt{ax + b}}{\ln^2 x }$
15	$y = \frac{\ln^2 x + b}{a\sqrt{x}}$	30	$y = \frac{1 + \sin^2 ax}{b^2 + x^2}$

Лабораторная работа 3. Создание разветвляющейся пользовательской функции

Цель работы: получение навыков по использованию условного оператора.

В редакторе VBA в том же модуле, где выполнялось предыдущее задание, создайте код функции с аргументом x в соответствии с заданным вариантом. Произведите отладку созданной функции с помощью встроенных функций MS Excel. Пример решения задачи изложен в подразделе 2.3 «Разветвляющиеся пользовательские функции».

Вариант 1

$$y = \begin{cases} 1/x, & \text{если } x \geq -5, x \neq 0; \\ x^2, & \text{если } x \leq -10; \\ \sqrt{|x+1|} & \text{в остальных случаях.} \end{cases}$$

(1)

(2)

(3)

Вариант 2

$$y = \begin{cases} x^2, & \text{если } x \leq 0, x \neq -10; \\ \sqrt{x+1}, & \text{если } x > 1; \\ 1/x & \text{в остальных случаях.} \end{cases}$$

(1)

(2)

(3)

Вариант 3

$$y = \begin{cases} x + e^{2x}, & \text{если } x \leq 0, x \neq -1; \\ \cos^2 x, & \text{если } 0 < x \leq 3,14; \\ x & \text{в остальных случаях.} \end{cases}$$

(1)

(2)

(3)

Вариант 4

$$y = \begin{cases} x^3, & \text{если } x > 10, x \neq 20; \\ x^2, & \text{если } -5 \leq x \leq 5; \\ \lg|x| & \text{в остальных случаях.} \end{cases}$$

(1)

(2)

(3)

Вариант 5

$$y = \begin{cases} \sqrt{x}, & \text{если } x \geq 100, x \neq 105; \\ \sqrt[3]{x}, & \text{если } x = 20 \text{ или } x = 40; \\ x^2 + 1 & \text{в остальных случаях.} \end{cases}$$

(1)

(2)

(3)

Вариант 6

$$y = \begin{cases} \sqrt[3]{x}, & \text{если } x > 20; \\ 1/x, & \text{если } x \leq 2 \text{ и } x \neq 0; \\ x^2 - 1 & \text{в остальных случаях.} \end{cases}$$

(1)

(2)

(3)

Вариант 7

$$y = \begin{cases} 8x + 1, & \text{если } x \geq 5, x \neq 9; \\ x^2 + |x|, & \text{если } x \leq 1; \\ x^3 + \sqrt{x} & \text{в остальных случаях.} \end{cases}$$

(1)

(2)

(3)

Вариант 8

$$y = \begin{cases} 1 - 3x, & \text{если } x > 0, x \neq 8; \\ x^2 - \sin x, & \text{если } x \leq -1; \\ \cos x & \text{в остальных случаях.} \end{cases}$$

(1)

(2)

(3)

Вариант 9

$$y = \begin{cases} x^3 + 1, & \text{если } x \geq 8, x \neq 10; \\ 2x^2 + \sqrt[3]{x}, & \text{если } x \leq 1; \\ \sqrt{x} & \text{в остальных случаях.} \end{cases}$$

(1)

(2)

(3)

Вариант 10

$$y = \begin{cases} \sqrt{x}, & \text{если } x \geq 4, x \neq 10; \\ 2x + 3, & \text{если } x \leq 1; \\ |x^3 - 4| & \text{в остальных случаях.} \end{cases}$$

(1)

(2)

(3)

Лабораторная работа 4. Организация ввода-вывода информации в среде VBA

Цель работы: получение навыков по использованию процедур ввода-вывода в диалоговые окна операторов обмена данными с ячейками рабочего листа Excel.

Задание 1. Наберите, сохраните в одном модуле и выполните три процедуры, текст которых приводится на рис. 52. Результаты покажите преподавателю. Основы решения задач изложены в разделе 3 «Основы программирования в среде VBA».

```
Sub privet()  
' Первая программа на VBA  
' Выполнил студент... (группа, Ф.И.О.)  
Dim Im As String, Gr As String  
Im = InputBox("Введи свое имя", "Ввод")  
Gr = InputBox("Введи свою группу", "Ввод")  
MsgBox "Привет " & Im & " из группы " & _  
      Gr, , "Приветствие"  
End Sub  
  
Sub prim1() ' Вывод данных в диалоговые окна  
Dim k As Integer, m As Integer, z As Integer  
k = InputBox("Введите значение k", "Ввод")  
m = InputBox("Введите значение m", "Ввод")  
MsgBox "k=" & k & " m=" & m, , "Исходные данные"  
z = m + k  
MsgBox "Z=" & z, , "Результаты"  
End Sub  
  
Sub prim2() ' Вывод данных в ячейки рабочего листа  
Dim k As Single, m As Single, z As Single  
Range("1:4").Clear  
k = InputBox("Введи k", "Ввод исходных данных")  
m = InputBox("Введи m", "Ввод исходных данных")  
Range("B1") = "Исходные данные"  
Cells(2, "B") = "k=" & k  
Cells(2, 3) = "m=" & m  
z = m + k  
Cells(3, 2) = "Результаты"  
Cells(4, 2) = "z=" & z  
End Sub
```

Рис. 52. Тексты процедур задания 1

Задание 2. Наберите, сохраните в том же модуле и отладьте процедуру, обеспечивающую с помощью встроенных окон диалог, схема которого приведена на рис. 53. Каждое окно сообщений снабдите соответствующим информационным значком. Проверьте работу каждой ветви программы. Пример решения задачи изложен в подразделе 3.4 «Типы данных».

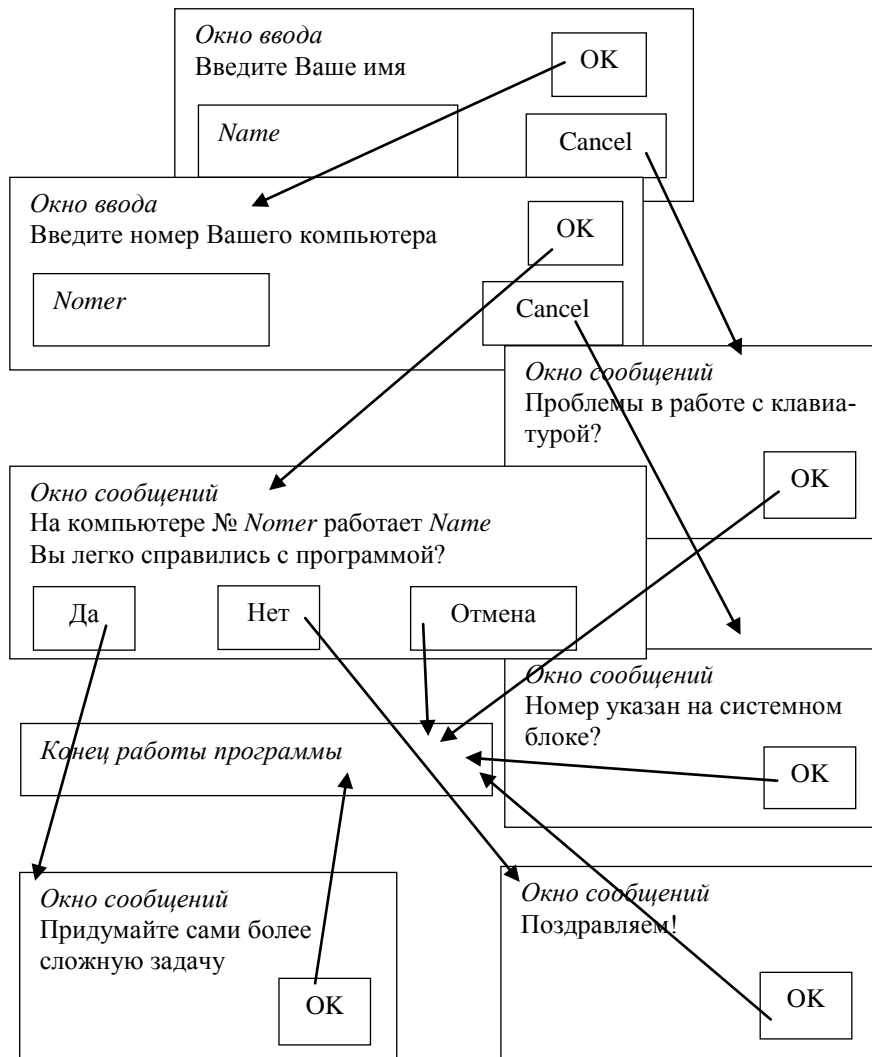


Рис. 53. Схема диалога для программы задания 2

Лабораторная работа 5. Программирование линейных алгоритмов в среде VBA

Цель работы: получение навыков реализации простых экономико-математических моделей.

Для указанного преподавателем варианта разработайте программу для вычисления значений экономических показателей. Данные для отладки подберите самостоятельно. Следует учесть, что интерфейс должен отражать экономическую сущность задачи. Для организации ввода, вывода исходных данных и результатов используйте следующие объекты:

- окно ввода и окно сообщений;
- ячейки рабочего листа;
- формы пользователя.

Основы решения задач изложены в разделе 3 «Основы программирования в среде VBA».

Вариант 1

Рассчитайте следующие показатели по отделению банка:

а) средний размер вклада на душу населения ($R^{вкл}$) по формуле

$$R^{вкл} = \frac{B}{S};$$

б) средний размер вклада на один лицевой счет ($R^{лсч}$) по следующей формуле:

$$R^{лсч} = \frac{B}{L};$$

в) число лицевых счетов в расчете на одного занятого ($Ч^{лсч}$) по формуле

$$Ч^{лсч} = \frac{L}{S^3};$$

г) долю занятых в народном хозяйстве в численности всего населения (D^3) по следующей формуле:

$$D^3 = \frac{S^3}{S},$$

где B – сумма остатка вклада, р.;

S – численность населения, чел.;

L – количество действующих лицевых счетов;

S^3 – численность занятых в народном хозяйстве, чел.

Вариант 2

Скорость обращения товарных запасов характеризуется оборачиваемостью в днях и коэффициентами оборачиваемости (прямым и обратным). Рассчитайте следующие коэффициенты:

а) оборачиваемость в днях (Le) по формуле

$$Le = \frac{E \cdot T}{N^p};$$

б) прямой коэффициент оборачиваемости ($K^п$) по следующей формуле:

$$K^п = \frac{N^p}{E};$$

в) обратный коэффициент оборачиваемости ($K^о$) по формуле

$$K^о = \frac{E}{N^p},$$

где E – средний остаток оборотных средств, р.;

T – длительность периода в днях (год – 360 дней, квартал – 90 и месяц – 30);

N^p – выручка от реализации продукции в оптовых ценах предприятия, р.

Вариант 3

Определите следующие показатели:

а) удельный вес стоимости машин и оборудования ($k^{маш}$) по формуле

$$k^{маш} = \frac{S^M}{S^O};$$

б) уровень годовой фондоотдачи с рубля основных фондов предприятия (λ) по следующей формуле:

$$\lambda = \lambda^O \cdot k^{маш},$$

где λ^O – фондоотдача с рубля стоимости рабочих машин и оборудования;
 S^M – стоимость машин и оборудования, р.;
 S^O – стоимость основных фондов, р.

Вариант 4

Рассчитайте следующие показатели:

а) коэффициент нарастания затрат ($k^{нар}$), вычисляемый по формуле

$$k^{нар} = \frac{C^M + 0,5C^O}{C};$$

б) среднюю продолжительность производственного цикла ($T^ц$) по предприятию в целом по следующей формуле:

$$T^ц = \frac{a \cdot T}{S \cdot k^{нар}},$$

где a – средний задел незавершенного производства, р.;
 T – длина анализируемого периода, дней;
 S – себестоимость выпущенной продукции, р.;
 C^M – затраты на материалы, р.;
 C^O – затраты на обработку, р.;
 C – затраты на производство, р.

Вариант 5

Определите следующие показатели:

а) величину приведенных затрат ($З^П$) на внедрение новой техники по формуле

$$З^П = З \cdot a_l;$$

б) затраты на внедрение новой техники ($З$) по следующей формуле:

$$З = C + E_n \cdot K,$$

где C – себестоимость единицы продукции, р.,
 E_n – нормативный коэффициент эффективности капитальных вложений;
 K – удельные капитальные вложения в новую технику, р.;
 a_l – коэффициент приведения, вычисляемый по формуле

$$a_l = (1 + E)^l,$$

где E – норматив приведения (0,1);
 l – количество лет, отделяющих затраты и результаты данного года от начала расчетного года.

Вариант 6

Определите следующие показатели:

а) фактическую экономию капитальных вложений (Δk) по формуле

$$\Delta k = (k_0 y_0 - k_1) A_1;$$

б) удельный вес годового объема продукции (y_0), производимой при использовании новой техники, в годовом объеме продукции, выпущенной на базовой технике по следующей формуле:

$$y_0 = \frac{A_1}{A_0},$$

где k_0 и k_1 – удельные капитальные вложения в базовую и новую технику, р.;
 A_0 и A_1 – годовые объемы продукции, производимой при использовании базовой и новой техники, в натуральных единицах.

Вариант 7

Определите следующие показатели:

а) сумму потерь от брака ($S_{\text{бп}}$) по формуле

$$S_{\text{бп}} = S^{\text{зп}} + R^{\text{ибп}} - W^{\text{бп}};$$

б) сумму возвращенной стоимости брака ($W_{\text{бп}}$) по следующей формуле:

$$W^{\text{бп}} = S_1 + S_2 + S_3,$$

где $S^{\text{зп}}$ – себестоимость забракованных изделий, р.;

$R^{\text{ибп}}$ – расходы по исправлению брака, р.;

S_1 – стоимость брака по цене использования;

S_2 – сумма удержаний с лиц-виновников брака;

S_3 – суммы, взысканные с поставщиков по претензиям за поставку недоброкачественных материалов.

Вариант 8

Определите следующие показатели:

а) полную себестоимость продукции (S) по формуле

$$S = Z + R^{\text{об}} + P^{\text{мз}} + R;$$

б) прочие расходы (R) по следующей формуле:

$$R = R^{\text{н}} + R^{\text{б}},$$

где Z – прямая заработная плата, р.;

$R^{\text{об}}$ – расходы на содержание и эксплуатацию оборудования, р.;

$P^{\text{мз}}$ – прямые материальные затраты, р.;

$R^{\text{н}}$ – накладные и внепроизводственные расходы, р.;

$R^{\text{б}}$ – непроизводственные расходы и потери от брака, р.

Вариант 9

Определите следующие показатели:

а) темп прироста производительности труда ($\Delta T_{\text{пр}}$) по формуле

$$\Delta T_{\text{пр}} = \frac{\varepsilon}{\chi + \varepsilon} \cdot 100;$$

б) темп роста ($\Delta T_{\text{р}}$) производительности труда по следующей формуле:

$$\Delta T_{\text{р}} = \Delta T_{\text{пр}} + 100,$$

где χ – расчетная (исходная) численность персонала, чел.;

s – среднесписочная численность персонала, чел.

ε – экономия численности персонала, чел. ($\varepsilon = s - \chi$).

Вариант 10

Определите следующие показатели:

а) уровень общей рентабельности (R) по формуле

$$R = \frac{P}{F + E^{\text{н}}},$$

б) расчетную прибыль (P) по следующей формуле:

$$P = P^{\text{б}} - \Pi_{\text{ф}} - \Pi_{\text{кр}} - \Pi_{\text{ппр}},$$

где F – среднегодовая стоимость основных производственных фондов, р.;

$E^{\text{н}}$ – средние остатки нормируемых оборотных средств, р.;

$P^{\text{б}}$ – балансовая прибыль, р.;

$\Pi_{\text{ф}}$ – плата за фонд, р.;

$\Pi_{\text{кр}}$ – проценты за кредит, р.;

$\Pi_{\text{ппр}}$ – прибыль, оставленная в распоряжении предприятия, р.

Лабораторная работа 6. Вычисление экономических показателей в среде VBA

Цель работы: получение навыков реализации экономико-математических моделей, содержащих альтернативные расчетные формулы.

Для указанного преподавателем варианта составьте графическую схему алгоритма и программу. Исходные данные для отладки программы подберите самостоятельно. Для организации ввода и вывода исходных данных и получения результатов используйте следующие объекты:

- стандартные диалоговые окна: окно ввода и окно сообщений (при выводе результатов предусмотрите использование информационных значков);
- ячейки рабочего листа (результаты, соответствующие различным диапазонам расчетного показателя, выведите с использованием разных элементов форматирования (размер, начертание, цвет шрифта, цвет фона), используя установку свойств объекта *Range*);
- формы пользователя.

Основы решения задач изложены в разделе 3 «Основы программирования в среде VBA».

Вариант 1

Для заданного расстояния до магазина рассчитайте стоимость транспортных расходов на обслуживание магазина при условии, что тариф зависит от расстояния следующим образом:

- если расстояние до магазина меньше 3 км, то тариф составляет 1500 р. за 1 км;
- если расстояние от 3 км (включительно) до 7 км – 1200 р. за 1 км;
- если расстояние не меньше 7 км – 1000 р. за 1 км.

Вариант 2

Для заданного времени разгрузки одного транспортного средства рассчитайте стоимость разгрузочных расходов базы при условии, что тариф зависит от времени разгрузки следующим образом:

- если время разгрузки не превышает 20 мин, то тариф равен 7000 р. за 1 мин;
- если время разгрузки больше 20 мин, но не превышает 50 мин, то тариф – 12000 р. за 1 мин;
- если время разгрузки больше 50 мин, то тариф – 15000 р. за 1 мин.

Вариант 3

Для заданного объема продажи и цены товара рассчитайте стоимость продажи товаров при условии, что величина скидки зависит от объема продажи следующим образом:

- при объеме продажи не более 10 шт. товара скидка не производится;
- при объеме продажи свыше 10 шт. и не более 50 шт. скидка составляет 5%;
- при объеме продажи свыше 50 шт. скидка составляет 10%.

Вариант 4

Для заданной продолжительности проживания в гостинице рассчитайте стоимость затрат на проживание при условии, что тариф проживания определяется следующим образом:

- если период проживания в гостинице меньше 7 дней, тариф равен 25000 р. за один день;
- если период находится в пределах от 7 дней (включительно) до 14 дней, то тариф – 20000 р. за день;
- если период не меньше 14 дней, тариф – 15000 р. за день.

Вариант 5

Для заданного объема закупок товара и величины торговой надбавки рассчитайте объем реализации товара за месяц как произведение объема закупок на коэффициент реализации при условии, что коэффициент реализации определяется следующим образом:

- если торговая надбавка не превышает 10%, то значение коэффициента реализации равно 0,9;
- если торговая надбавка находится в пределах от 10% до 20% (включительно), то коэффициент – 0,75;
- если торговая надбавка больше 20 %, то коэффициент – 0,5.

Вариант 6

Для заданного объема произведенной партии продукции и цены 1 т сырья рассчитайте объем и стоимость используемого сырья. Объем требуемого сырья вычисляется как произведение объема произведенной партии продукции на норматив расхода сырья. Норма расхода сырья определяется следующим образом:

- при производстве продукции объемом менее 140 т расходуется 1,4 т сырья на 1 т продукции;
- при производстве продукции объемом от 140 (включительно) до 200 т расходуется 1,3 т сырья на 1 т продукции;
- при производстве продукции объемом не менее 200 т расходуется 1,2 т сырья на 1 т продукции.

Вариант 7

Заданы величины объема продажи и цены товара. Вычислите стоимость продажи как произведение объема продажи на отпускную цену. Отпускная цена рассчитывается как сумма цены товара и величины торговой надбавки в рублях. Величина торговой надбавки зависит от объема продажи следующим образом:

- при объеме продажи не более 40 т – 18%;
- при объеме продажи от 40 т до 60 т включительно – 10%;
- при объеме продажи свыше 60 т – 5%.

Вариант 8

Определите стоимость билета для пассажиров, едущих на заданное расстояние. Стоимость проезда в транспорте зависит от расстояния, которое проезжает пассажир, и тарифа за 1 км пути, который устанавливается в зависимости от дальности следующим образом:

- если расстояние меньше 10 км, то тариф проезда равен 500 р. за 1 км;
- если расстояние находится в пределах от 10 км до 100 км, то тариф проезда равен 400 р. за 1 км;
- если расстояние не меньше 100 км, то тариф проезда равен 300 р. за 1 км.

Вариант 9

Заданы следующие значения: оклад, стаж, возраст, пол, количество детей. Вычислите размер надбавки за качество в соответствии со следующими правилами:

- надбавка начисляется в размере 50% от оклада всем сотрудникам, имеющим стаж работы больше 20 лет, и тем сотрудникам с меньшим стажем работы, у которых на иждивении имеется не меньше трех детей;
- надбавка начисляется в размере 30% от оклада сотрудникам со стажем работы больше 15 лет (но не больше 20 лет) и сотрудникам с меньшим стажем, имеющим на иждивении двух детей;
- надбавка начисляется в размере 20% от оклада сотрудникам со стажем работы больше 10 лет (но не больше 15%) при условии, что они достигли пенсионного возраста (55 лет для женщин и 60 лет для мужчин);
- остальным сотрудникам надбавка не назначается.

Вариант 10

Оцените скорость оборота денежных средств (S), вычисляемую по формуле

$$S = \frac{E \cdot T}{N},$$

где E – средний остаток оборотных средств, р.;

T – длительность периода в днях (год – 360, квартал – 90 и месяц – 30 дней);

N – выручка от реализации продукции в оптовых ценах предприятия, р.

Результат оценки определяется следующими условиями:

- $S \geq 20$ – низкий уровень оборачиваемости;
- $S \in [10; 20)$ – средний уровень оборачиваемости;
- $S < 10$ – высокий уровень оборачиваемости.

Вариант 11

Оцените уровень (d) заработной платы (Z) в полной себестоимости продукции (U) по формуле

$$U = Z + R^{об} + P + R; \quad d = \frac{Z}{U},$$

где Z – прямая заработная плата, р.;

$R^{об}$ – расходы на содержание и эксплуатацию оборудования, р.;

P – прямые материальные затраты, р.;

R – расходы, р.

Результат оценки определяется следующими условиями:

- $d \geq 0,8$ – высокий уровень заработной платы;
- $d \in [0,6; 0,8)$ – средний уровень заработной платы;
- $d < 0,6$ – низкий уровень заработной платы.

Вариант 12

Оцените уровень (d) потерь от брака ($R^{бп}$) в общей сумме расходов (R) следующим образом:

$$R = R^{нак} + R^{непр} + R^{бп}; \quad d = \frac{R^{бп}}{R},$$

где $R^{нак}$ – накладные внепроизводственные расходы, р.;
 $R^{непр}$ – непроизводственные расходы, р.;
 $R^{бп}$ – потери от брака, р.

Результат оценки определяется следующими условиями:

- $d > 0,5$ – недопустимо высокий уровень потерь от брака;
- $d \in (0,3; 0,5)$ – высокий уровень потерь от брака;
- $d \leq 0,3$ – допустимый уровень потерь от брака.

Вариант 13

Оцените уровень (d) расходов по исправлению брака (R) в сумме потерь от брака (P) следующим образом:

$$P = S + R - W; \quad d = \frac{R}{P},$$

где S – себестоимость забракованных изделий, р.;
 R – расходы по исправлению брака, р.;
 W – сумма возвращенной стоимости брака, р.

Результат оценки определяется следующими условиями:

- $d > 0,8$ – высокий уровень расходов по исправлению брака;
- $d \in [0,6; 0,8]$ – допустимый уровень расходов по исправлению брака;
- $d < 0,6$ – средний уровень расходов по исправлению брака.

Вариант 14

Оцените уровень (N^p) товарной продукции за отчетный период (N^T) в полном объеме реализации продукции (N) с использованием следующих формул:

$$N^p = \frac{N^T}{N}; \quad N = N^T + N^O + N^S,$$

где N – полный объем реализации продукции, р.;
 N^T – реализация товарной продукции за отчетный период, р.;
 N^O – остатки отгруженной продукции, р.;
 N^S – остатки готовой продукции на складе, р.

Результат оценки определяется следующими условиями:

- $N^p > 0,85$ – высокий уровень товарной продукции;
- $N^p \in (0,7; 0,85]$ – средний уровень товарной продукции;
- $N^p \leq 0,75$ – низкий уровень товарной продукции.

Вариант 15

Оцените уровень (U) расходов по сбыту (R^s) в общей сумме расходов (R) с использованием следующих формул:

$$U = \frac{R^s}{R}; \quad R = R^{себ} + R^s + R^a + R^{np},$$

где R – общая сумма расходов, р.;
 $R^{себ}$ – себестоимость реализованной продукции, р.;
 R^s – расходы по сбыту, р.;
 R^a – административные расходы, р.;
 R^{np} – прочие расходы, р.

Результат оценки определяется следующими условиями:

- $U > 0,5$ – недопустимо высокие расходы;
- $U \in (0,2; 0,5]$ – высокие расходы;
- $U \leq 0,2$ – допустимый уровень расходов.

Вариант 16

Оцените долю (P) расходов в семейном бюджете на питание (R^{nut}) в общем объеме расходов (R) с использованием следующих формул:

$$P = \frac{R^{nut}}{R}; \quad R = R^{nut} + R^{ком} + R^{од} + R^{пр},$$

где R – расходы семьи, р.;

R^{nut} – расходы на питание, р.;

$R^{ком}$ – оплата жилья и коммунальных услуг, р.;

$R^{од}$ – расходы на одежду и обувь, р.;

$R^{пр}$ – прочие расходы, р.

Результат оценки определяется следующими условиями:

- $P > 0,7$ – высокая доля расходов;
- $P \in [0,3; 0,7]$ – средняя доля расходов;
- $P < 0,3$ – низкая доля расходов.

Вариант 17

Оцените уровень (P) транспортных расходов (R^{mp}) в общей сумме расходов предприятия (R) с использованием следующих формул:

$$P = \frac{R^{mp}}{R}; \quad R = R^{себ} + R^{сб} + R^{mp} + R^{накл},$$

где R – сумма расходов предприятия, р.;

$R^{себ}$ – себестоимость реализованной продукции, р.;

$R^{сб}$ – расходы по сбыту, р.;

R^{mp} – транспортные расходы, р.;

$R^{накл}$ – накладные расходы, р.

Результат оценки определяется следующими условиями:

- $P > 0,3$ – высокий уровень транспортных расходов;
- $P \in [0,1; 0,3]$ – средний уровень транспортных расходов;
- $P < 0,1$ – низкий уровень транспортных расходов.

Лабораторная работа 7. Табулирование функций в среде VBA

Цель работы: получение навыков использования операторов цикла.

Для указанного преподавателем варианта из табл. 14 составьте и произведите отладку процедуры вычисления значений функции $y = f(x, a, b)$ для значений аргумента x , изменяющегося в интервале от $x_{нач}$ до $x_{кон}$ с шагом Δx , и заданных констант a и b .

Исходные данные ($x_{нач}$, $x_{кон}$, Δx , a , b) для отладки процедуры выберите самостоятельно из интервала значений, где заданные функции определены. Процедура должна обеспечивать вывод результатов (значения аргумента x и функции y) в последовательность окон сообщений (в одном окне одна пара значений x и y).

После приема работы преподавателем преобразуйте процедуру так, чтобы вывод результатов происходил в одно окно сообщений. Создайте процедуру табулирования этой же функции с выводом результатов в ячейки рабочего листа Excel.

Примеры решения задач приведены в подразделе 4.2 «Табулирование функции».

Лабораторная работа 8. Вычисления в одномерных числовых массивах

Цель работы: получение навыков реализации типовых алгоритмов обработки числовых массивов.

Для варианта, указанного преподавателем, разработайте процедуру и произведите ее отладку для самостоятельно подготовленных тестов. Процедура должна использовать диалоговые окна для ввода размерности и элементов массива, ввода некоторых дополнительных чисел и вывода результатов вычислений. Отладка должна быть проведена для полного набора тестов, отражающего всевозможные варианты выполнения вычислений.

Примеры решения задач приведены в подразделе 4.5 «Простые операции над одномерными массивами».

Вариант 1

Найдите количество чисел, принадлежащих промежутку $[A, B]$, и сумму чисел, стоящих на местах, кратных 3.

Вариант 2

Найдите сумму чисел, меньших заданного D , и количество чисел, стоящих на четных местах и больших заданного C .

Вариант 3

Найдите произведение всех чисел, стоящих на местах, кратных 4, и количество чисел, не больших заданного A .

Вариант 4

Найдите количество чисел, меньших заданного X , и произведение всех отрицательных чисел, стоящих на нечетных местах.

Вариант 5

Найдите количество чисел, не принадлежащих промежутку $(X, Y]$, и сумму отрицательных чисел, стоящих на четных местах.

Вариант 6

Найдите количество неотрицательных чисел и определите сумму чисел, стоящих на местах, кратных 3 и неравных заданному F .

Вариант 7

Вычислите произведение чисел, принадлежащих промежутку $(A, B]$, и количество отрицательных чисел, стоящих на местах, кратных 3.

Вариант 8

Найдите количество нулей во всем массиве и определите сумму квадратов чисел, принадлежащих промежутку (A, B) и стоящих на местах, кратных 4.

Вариант 9

Найдите среднее арифметическое отрицательных чисел и определите количество чисел, по величине больших A и стоящих на четных местах.

Вариант 10

Найдите среднее арифметическое положительных чисел, стоящих на нечетных местах, и количество чисел, меньших заданного B .

Вариант 11

Найдите среднее арифметическое чисел, принадлежащих промежутку $[A, B)$, и количество положительных чисел, стоящих на местах, кратных 4.

Вариант 12

Найдите среднее арифметическое чисел, неравных заданному C , и произведение неположительных чисел, стоящих на четных местах.

Вариант 13

Найдите среднее арифметическое чисел, больших заданного D и стоящих на нечетных местах, определите количество чисел, не больших заданного F .

Вариант 14

Найдите среднее арифметическое чисел, не попадающих в промежуток $[A, B]$, и количество положительных чисел, стоящих на местах, кратных 3.

Вариант 15

Найдите среднее арифметическое ненулевых чисел и количество чисел, по величине не больших A и стоящих на четных местах.

Вариант 16

Найдите среднее арифметическое положительных чисел, стоящих на нечетных местах, и произведение чисел, меньших заданного C .

Лабораторная работа 9.

Перестановки элементов в одномерных числовых массивах

Цель работы: получение навыков реализации процедур по перестановке элементов в одномерных массивах.

Для варианта, указанного преподавателем, разработайте процедуру и произведите ее отладку для самостоятельно подготовленных тестов. Процедура должна использовать диалоговые окна для ввода размерности и элементов массива, определить значения и местоположения минимального или максимального элемента и выводить элементы полученного массива. Отладка должна быть проведена для полного набора тестов, отражающего различные варианты расположения искоемых элементов в массиве.

Примеры решения задач приведены в подразделе 4.6 «Определение минимального и максимального элементов массива».

Вариант 1

Найдите первый максимальный элемент и поменяйте его местами с пятым элементом массива.

Вариант 2

Найдите последний минимальный элемент и поменяйте его местами с предыдущим элементом массива.

Вариант 3

Найдите первый минимальный элемент и поменяйте его местами с последующим элементом массива.

Вариант 4

Найдите первый максимальный элемент и поменяйте его местами с шестым элементом массива.

Вариант 5

Найдите последний минимальный элемент и поменяйте его местами с третьим элементом массива.

Вариант 6

Найдите первый минимальный элемент и поменяйте его местами с третьим элементом массива.

Вариант 7

Найдите последний максимальный элемент и поменяйте его местами с четвертым элементом массива.

Вариант 8

Найдите первый максимальный элемент и поменяйте его местами с предыдущим элементом массива.

Вариант 9

Найдите первый максимальный элемент и поменяйте его местами с последним минимальным элементом массива.

Вариант 10

Найдите последний максимальный элемент и поменяйте его местами с первым минимальным элементом массива.

Лабораторная работа 10.

Формирование новых массивов

Цель работы: получение навыков по реализации процедур формирования новых массивов.

Для варианта, указанного преподавателем, разработайте процедуру и произведите ее отладку для самостоятельно подготовленных тестов. Процедура должна использовать диалоговые окна для ввода двух целочисленных массивов, формирование нового массива и вывод полученного массива. Учтите, что исходные массивы могут быть разной длины.

Примеры решения задач приведены в подразделе 4.7 «Формирование массива».

Вариант 1

Сформируйте массив из элементов исходных массивов, больших второго элемента первого массива, и положительных элементов второго массива.

Вариант 2

Сформируйте массив из отрицательных элементов первого массива и элементов обоих массивов, больших первого элемента второго массива.

Вариант 3

Сформируйте массив из положительных элементов первого массива и элементов обоих массивов, больших последнего элемента второго массива.

Вариант 4

Сформируйте массив из элементов исходных массивов, меньших произведения последних элементов заданных массивов, и отрицательных элементов второго массива.

Вариант 5

Сформируйте массив из элементов исходных массивов, не превышающих первого элемента первого массива, и элементов второго массива, больших заданного числа.

Вариант 6

Сформируйте массив из элементов исходных массивов, не превышающих третий элемент каждого из них соответственно, и положительных элементов первого массива.

Вариант 7

Сформируйте массив из элементов исходных массивов, не превышающих сумму первых элементов исходных массивов, и элементов первого массива, больших заданного числа.

Вариант 8

Сформируйте массив из положительных элементов исходных массивов, меньших 10, и отрицательных элементов второго массива.

Вариант 9

Сформируйте массив из нечетных элементов первого массива, больших 5, и элементов обоих массивов, не меньших заданного числа.

Вариант 10

Сформируйте массив из отрицательных элементов исходных массивов и четных элементов второго массива, больших заданного числа.

Вариант 11

Сформируйте массив из элементов исходных массивов, больших первого элемента второго массива, и нечетных элементов второго массива.

Вариант 12

Сформируйте массив из отрицательных четных элементов первого массива и положительных элементов исходных массивов, не меньших заданного числа.

Вариант 13

Сформируйте массив из нечетных элементов первого массива и элементов исходных массивов, не больших 3.

Вариант 14

Сформируйте массив из нечетных элементов второго массива, больших последнего элемента второго массива, и элементов исходных массивов, меньших последнего элемента первого массива.

Вариант 15

Сформируйте массив из тех элементов исходных массивов, которые меньше заданного числа, и нечетных элементов второго массива, больших последнего элемента первого массива.

Вариант 16

Сформируйте массив из элементов исходных массивов, принадлежащих промежутку $[-4; 6]$, и четных элементов первого массива, больших 12.

**Лабораторная работа 11.
Обработка экономической информации
на базе числовых массивов**

Цель работы: получение навыков обработки экономической информации на базе числовых массивов.

Для варианта, указанного преподавателем, разработайте процедуру и произведите ее отладку для самостоятельно подготовленных тестов. Процедура должна использовать диалоговые окна для ввода размерности и элементов массива, вычислять требуемые показатели и выводить полученные результаты.

Вариант 1

По данным о ежемесячных материальных затратах организации за отчетный период определите следующее:

- сколько месяцев организация работала неэффективно, т. е. превысила отраслевой норматив;
- какова доля затрат этих месяцев в общей сумме материальных затрат организации за отчетный период.

Вариант 2

По данным о ежемесячном объеме выпуска товарной продукции предприятием за отчетный период определите следующее:

- количество месяцев, в которых объем выпуска продукции организацией был меньше среднемесячного уровня, и количество месяцев, в которых предприятие работало успешно;
- какова доля объема продукции за месяцы, в которые предприятие работало успешно, в общем объеме выпущенной продукции организацией за отчетный период.

Вариант 3

По данным о ежемесячном объеме выплаченной заработной платы работникам организации за отчетный период определите следующее:

- среднемесячный размер фонда заработной платы;
- долю суммы заработной платы, выплаченной в первые два месяца анализируемого периода, в общей сумме выплат за весь период.

Вариант 4

По данным о ежемесячно получаемой прибыли организацией в течение отчетного периода определите следующее:

- количество месяцев, в которых объем получаемой прибыли был выше среднего уровня;
- какова доля прибыли этих месяцев в общей сумме прибыли, полученной организацией за анализируемый период.

Вариант 5

По данным о ежемесячных материальных затратах организации за отчетный период определите следующее:

- количество месяцев, когда организация работала лучше, чем предусмотрено нормативом;
- какова доля материальных затрат этих месяцев в общей сумме материальных затрат организации за отчетный период.

Вариант 6

По данным о среднемесячной численности персонала фирмы за отчетный период определите следующее:

- количество месяцев, когда численность персонала фирмы была меньше планового значения;
- удельный вес этих месяцев в отчетном периоде.

Вариант 7

По данным о месячных объемах фактически привлеченных средств во вклады филиалами банка за отчетный период рассчитайте следующее:

- общую сумму привлеченных филиалами банка средств;
- среднюю сумму привлеченных средств за те месяцы, когда привлекалось средств меньше заданного значения.

Вариант 8

По данным о ежемесячных затратах организации на сырье за отчетный период определите следующее:

- сколько месяцев организация расходовала сырья больше, чем предусмотрено нормативом;
- какова доля этих затрат в общей сумме затрат на сырье организации за анализируемый период.

Вариант 9

По данным о цеховых расходах организации за конкретный месяц определите следующее:

- количество цехов, работавших со среднецеховым уровнем расходов;
- долю расходов тех цехов организации, в которых уровень расходов был выше расходов первого цеха.

Вариант 10

По данным о фонде заработной платы производственных рабочих цехов организации найдите следующее:

- среднецеховую заработную плату производственных рабочих;
- долю заработной платы рабочих тех цехов, которые имеют фонд заработной платы ниже заданного значения.

Лабораторная работа 12. Табулирование функции двух переменных

Цель работы: получение навыков составления вложенных циклов.

Для варианта функции вида $Z = f(x, y)$ из табл. 16, составьте процедуру, выводящую таблицу значений при изменении аргумента x в пределах от $x_{нач}$ до $x_{кон}$ с шагом Δx и аргумента y в пределах от $y_{нач}$ до $y_{кон}$ с шагом Δy . Значения пределов изменения аргументов x и y выберите самостоятельно.

Таблица 16. Варианты заданий по табулированию функции двух аргументов

Вариант	$Z = f(x, y)$
1	$Z = \sin x + \cos y$
2	$Z = \cos x + \sin y$
3	$Z = \sin x + \cos y^2$
4	$Z = \sin^2 x + \cos y^2$
5	$Z = \sqrt{x} + \ln y$
6	$Z = \sin x + \ln y$
7	$Z = \cos x + \lg y$
8	$Z = \lg x + \lg y$
9	$Z = \ln x + \lg y$
10	$Z = \lg x + \lg y$

Лабораторная работа 13. Вычисления в двумерных массивах

Цель работы: получение навыков обработки двумерных массивов.

Задана матрица размером $N \times M$. Составьте процедуру, которая должна выполнять следующие действия:

- 1) ввод размерности и элементов матрицы (размером 4×5 или 5×4) и некоторых дополнительных чисел;
- 2) операции над элементами матрицы в соответствии с вариантом, указанным преподавателем;
- 3) вывод исходных данных и результатов вычислений.

Исходные данные для отладки программы выберите самостоятельно.

Вариант 1

Вычислите сумму положительных элементов, расположенных в столбцах с четными номерами.

Вариант 2

Вычислите произведение отрицательных элементов, расположенных в строках с нечетными номерами.

Вариант 3

Вычислите сумму квадратов элементов из интервала $[A, B]$ и расположенных в строках с четными номерами.

Вариант 4

Определите количество элементов, больших заданного A и расположенных в строках с нечетными номерами.

Вариант 5

Вычислите сумму элементов, меньших заданного B и расположенных в столбцах с номерами, кратными 3.

Вариант 6

Вычислите произведение положительных элементов, расположенных в строках с номерами, кратными 4.

Вариант 7

Вычислите сумму квадратов отрицательных элементов, расположенных в столбцах с четными номерами.

Вариант 8

Определите количество элементов, не принадлежащих промежутку (A, B) и расположенных в столбцах с нечетными номерами.

Вариант 9

Вычислите сумму элементов, не меньших заданного D и расположенных в строках с четными номерами.

Вариант 10

Вычислите произведение элементов, не больших заданного F и расположенных в строках с нечетными номерами.

**Лабораторная работа 14.
Поиск элементов в двумерных массивах**

Цель работы: получение навыков обработки двумерных массивов.

Для варианта, указанного преподавателем, составьте процедуру, выполняющую действия, аналогичные перечисленным в лабораторной работе 13.

Вариант 1

Среди элементов, расположенных выше и ниже главной диагонали матрицы, найдите минимальные элементы и вычислите произведение этих элементов.

Вариант 2

В каждом столбце матрицы найдите произведение положительных элементов и вычислите сумму этих произведений.

Вариант 3

В каждом столбце матрицы найдите произведение положительных элементов и вычислите сумму этих произведений.

Вариант 4

Определите, где больше чисел, равных заданному числу Z: выше главной диагонали или ниже ее.

Вариант 5

В каждом столбце матрицы найдите сумму отрицательных элементов и вычислите произведение этих сумм.

Вариант 6

В каждой строке матрицы найдите самый правый положительный элемент и вычислите сумму этих элементов.

Вариант 7

В каждой четной по номеру строке матрицы найдите максимальный элемент и вычислите сумму этих элементов.

Вариант 8

В каждом четном столбце матрицы найдите минимальный элемент и вычислите произведение этих элементов.

Вариант 9

В каждой нечетной строке матрицы найдите произведение отрицательных элементов и вычислите сумму этих произведений.

Вариант 10

Определите, в какой части матрицы больше элементов, значения которых попадают в интервал [0, 10]: выше побочной диагонали или ниже ее.

**Лабораторная работа 15.
Циклы с досрочным завершением**

Цель работы: получение навыков составления вложенных циклов с предусловиями.

Для варианта, указанного преподавателем, составьте процедуру, которая выполняет поиск в прямоугольной матрице элементов, соответствующих заданному условию. При этом поиск надо организовать таким образом, чтобы были исключены лишние проверки. Результаты поиска выводите в окно сообщений, а все исходные данные задавайте в ячейках листа рабочей книги. Исходные данные для отладки программы выберите самостоятельно.

Вариант 1

Определите, есть ли в матрице строка, состоящая только из положительных элементов.

Вариант 2

Определите количество строк матрицы, в которых нет положительных элементов.

Вариант 3

Определите, есть ли в матрице строка, сумма элементов которой равна заданному значению.

Вариант 4

Определите, есть ли в матрице столбец, состоящий только из отрицательных элементов.

Вариант 5

Определите, есть ли в матрице строка, в которой все элементы равны между собой.

Вариант 6

Определите, есть ли в матрице строка, в которой нет нулевых элементов.

Вариант 7

Определите номер первого столбца матрицы, в котором произведение элементов отрицательное.

Вариант 8

Определите, есть ли в матрице строка, в которой на первом месте стоит максимальный элемент строки.

Вариант 9

Укажите, есть ли в матрице столбец, в котором на последнем месте стоит минимальный элемент столбца.

Вариант 10

Определите номер первой строки матрицы, в которой больше трех положительных элементов.

**Лабораторная работа 16.
Поиск в символьных данных и их модификация**

Цель работы: получение навыков обработки символьной информации.

Задана строка символов. Составьте процедуру, которая должна выполнять следующие действия:

- 1) ввод строки длиной не меньше двадцати символов;
- 2) операции над символами строки в соответствии с вариантом, указанным преподавателем;
- 3) вывод исходных данных и результатов вычислений.

Исходные данные для отладки процедуры выберите самостоятельно.

Вариант 1

После каждого символа «,» вставьте пробел и подсчитайте количество символов «А» и «В» отдельно.

Вариант 2

Замените символ «*» на символы «++» и подсчитайте общее количество символов «F» и «D».

Вариант 3

Удалите каждую пару символов «PQ» и подсчитайте общее количество символов «.» и «,» в строке.

Вариант 4

Подсчитайте количество пар символов «+-» и замените каждый символ «*» на символы «/-».

Вариант 5

Удалите каждый символ «А», стоящий после символа «,» и подсчитайте количество пар «BC» и «DE» отдельно.

Вариант 6

Удалите каждый символ «?», стоящий после символа «;», и подсчитайте общее количество символов «0» и «1».

Вариант 7

Подсчитайте количество символов «+», стоящих между символами «А» и «В», и замените каждый символ «0» на «ОО».

Вариант 8

В середину каждой пары символов «АВ» вставьте символ «*» и подсчитайте, сколько раз в строке символ «I» стоит перед «2».

Вариант 9

Удалите все символы «,» из строки и подсчитайте количество символов «F», стоящих после символа «+», и количество символов «F», стоящих после символа «-» (отдельно).

Вариант 10

Удалите из строки все символы «,» и «.», подсчитайте общее количество символов «X» и «Y», стоящих после символа «*».

Лабораторная работа 17.
Символьные данные: работа со словами

Цель работы: получение навыков выделения слов из символьной информации.

Задана строка символов, состоящая из слов, разделенных одним или несколькими пробелами. Составьте процедуру, которая должна выполнять следующие действия:

- 1) ввод строки, в которой не меньше четырех слов, отделенных друг от друга разным количеством пробелов;
- 2) операции над фрагментами строки в соответствии с вариантом, указанным преподавателем;
- 3) вывод исходных данных и результатов работы программы.

Вариант 1

Подсчитайте количество слов и после каждого поставьте запятую.

Вариант 2

Подсчитайте количество букв в третьем слове.

Вариант 3

Определите количество слов, начинающихся с символа «А».

Вариант 4

Для каждого слова укажите количество букв, из которых оно состоит.

Вариант 5

Удалите последнюю букву в каждом слове.

Вариант 6

Подсчитайте количество символов «О» в предпоследнем слове.

Вариант 7

Определите количество слов, которые заканчиваются символом «Е».

Вариант 8

Определите количество слов, которые начинаются символом «Е».

Вариант 9

Определите количество слов, в которых нет ни одного символа «Е».

Вариант 10

Каждое слово заключите в кавычки.

Лабораторная работа 18. Использование форм при программировании линейных алгоритмов

Цель работы: получение навыков составления процедур для обработки событий элементов управления формы.

В соответствии с вариантом лабораторной работы 5 составьте программу для расчета экономических показателей на основе пользовательской формы для ввода исходных данных (см. пример в подразделе 7.4 «Использование форм для организации диалога с пользователем»). На форме необходимо разместить элементы управления, указывающие место вывода исходных данных и результатов.

Лабораторная работа 19. Использование форм при программировании разветвляющихся алгоритмов

Цель работы: получение навыков использования элементов управления формы для вывода результатов.

В соответствии с вариантом лабораторной работы 6 составьте программу для расчета экономических показателей на основе пользовательской формы для ввода исходных данных и вывода результатов (см. пример в подразделе 7.4 «Использование форм для организации диалога с пользователем»).

Лабораторная работа 20. Использование форм при обработке экономических показателей на базе числовых массивов

Цель работы: расширение навыков использования элементов управления формы.

В соответствии с вариантом лабораторной работы 11 составьте программу для расчета экономических показателей на основе одной пользовательской формы.

ВОПРОСЫ ДЛЯ ПОДГОТОВКИ К ТЕСТИРОВАНИЮ И ЭКЗАМЕНУ ПО КУРСУ «ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ» ДЛЯ СТУДЕНТОВ ЗАОЧНОЙ ФОРМЫ ОБУЧЕНИЯ

1. *Введение в технологии программирования в среде Visual Basic for Applications (VBA):*

1.1. Общая характеристика VBA: общая языковая платформа для всех приложений пакета MS Office, интегрированная среда разработки программ, интерпретация программ, запуск редактора VBA.

1.2. Основные понятия VBA: объект, свойство, событие, метод, процедура, модуль, проект, макрос.

2. *Использование макросов:*

2.1. Основные понятия макроса: имя макроса, автоматическая запись макроса, ввод текста процедур на макроязыке, описание макроса.

2.2. Этапы автоматической записи макроса: начало записи, ввод последовательности команд, остановка записи.

2.3. Отличие макроса с абсолютными ссылками от макроса с относительными ссылками.

2.4. Способы удаления, редактирования, переименования макроса и запуска макроса на выполнение.

2.5. Технология назначения макросу клавиш быстрого реагирования.

2.6. Технология назначения макроса графическому объекту.

2.7. Технология назначения макроса стандартной кнопке.

2.8. Технология назначения макроса кнопке панели управления.

2.9. Технология создания панели управления пользователя.

3. *Редактор VBA:*

3.1. Назначение окна проекта.

3.2. Использование окна свойств.

3.3. Использование окна модуля.

3.4. Работа с формами.

3.5. Задание параметров редактора VBA.

4. *Пользовательские функции:*

4.1. Общие правила набора пользовательских функций.

- 4.2. Правила набора математических пользовательских функций с использованием стандартных математических функций VBA.
- 4.3. Технология удаления функции, определенной пользователем.
- 4.4. Правила набора разветвляющихся пользовательских функций.
5. *Основы программирования на VBA:*
 - 5.1. Структура простой программы.
 - 5.2. Допустимые имена (идентификаторы).
 - 5.3. Используемые типы данных.
 - 5.4. Правила описания переменных.
 - 5.5. Правила записи констант.
 - 5.6. Используемые арифметические операторы.
 - 5.7. Оператор конкатенации.
 - 5.8. Операторы сравнения.
 - 5.9. Используемые логические операторы.
 - 5.10. Приоритеты операций в выражениях.
 - 5.11. Правила записи и использования оператора присваивания.
 - 5.12. Правила оформления кода.
 - 5.13. Использование окна ввода.
 - 5.14. Использование окна сообщений.
 - 5.15. Ввод данных с рабочего листа и вывод результатов в ячейки листа Excel.
6. *Технологии программирования простых циклических алгоритмов в VBA:*
 - 6.1. Правила записи и выполнения оператора цикла While – Wend.
 - 6.2. Правила записи и выполнения оператора цикла For – Next.
 - 6.3. Технология реализации табулирования функции с выводом результатов в последовательность окон сообщений.
 - 6.4. Технология реализации табулирования функции с выводом результатов в одно окно сообщений.
 - 6.5. Технология реализации табулирования функции с выводом результатов в ячейки рабочего листа Excel.
 - 6.6. Правила описания массивов.
 - 6.7. Организация ввода и вывода элементов массива.
 - 6.8. Простые операции над одномерными массивами (суммирование элементов, подсчет количества и др.).
 - 6.9. Определение минимального и максимального элементов массива.
 - 6.10. Формирование нового массива из некоторых элементов исходного.
7. *Обработка двумерных массивов:*
 - 7.1. Правила описания двумерных массивов.
 - 7.2. Организация ввода и вывода двумерных массивов.
 - 7.3. Примеры обработки двумерных массивов.

ПРИМЕРНЫЕ ТЕСТОВЫЕ ЗАДАНИЯ

Закончите фразы, выбрав один или несколько правильных ответов.

1. В процессе автоматической записи макроса:
 - а) любое нажатие клавиши или щелчок мыши автоматически фиксируется в тексте процедуры;
 - б) перед каждым нажатием клавиши надо выделить соответствующую ячейку рабочего листа;
 - в) в текст процедуры заносятся только вводимые с клавиатуры значения.
2. Включение процедуры записи макроса осуществляется при выполнении команды:
 - а) *Сервис\Макрос\Редактор Visual Basic*;
 - б) *Сервис\Макрос\Начать запись*;
 - в) *Сервис\Макрос\Редактор сценариев*.
3. Выключить запись макроса можно:
 - а) с помощью кнопки *Остановить запись* на панели *Остановка записи*;
 - б) с помощью команды *Сервис\Макрос\Остановить запись*;
 - в) нажатием клавиши *Esc*.
4. Пользовательская функция

Function F1(x)

$$F1 = \cos(\sqrt{x})$$

End Function

вычисляет значение следующей функции:

а) $y = \cos\sqrt{x}$;

б) $y = \cos^2 x$;

в) $y = \cos x^2$.

5. Пользовательская функция

Function F2(x)

If (x>=7) or (x=2) then

F2=sqr(x)

Else

If x<=0 then

F2= x^2

Else

*F2= x*2*

End if

End if

End Function

при x=-9 возвращает следующее значение:

а) 81;

б) 3;

в) 18.

6. Нижеприведенной программе соответствует следующее окно вывода:

Sub Табуляция2()

Dim x As Single, xn As Single, xk As Single, dx As Single, y As Single

xn = InputBox("Введите начальное значение x", "Табуляция")

xk = InputBox("Введите конечное значение x", "Табуляция")

dx = InputBox("Введите шаг изменения x", "Табуляция")

s = " x " & " y "

x = xn

While x < (xk + dx / 1000)

*y = 2 * x*

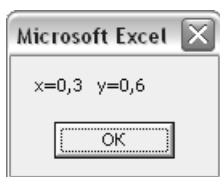
s = s & Chr(13) & x & " " & y

x = x + dx

Wend

MsgBox s

End Sub



а)



б)

	A	B	C
1	N	x	y
2	1	0,3	0,6
3	2	0,7	1,4

в)

7. Нижеприведенная программа решает следующую задачу:

Option Explicit

Option Base 1

Sub ВводВыводМассива()

Dim A(10) As Single

Dim I As Integer

Dim N As Integer

Dim C As String

C = "Исходный массив"

N = InputBox("Количество элементов массива?")

For I = 1 To N

A(I) = InputBox("A(" & I & ")=?")

C = C & vbCr & I & vbTab & A(I)

Next I

MsgBox C

End Sub

- а) ввод элементов одномерного целочисленного массива и вывод введенных значений в последовательность окон вывода;
- б) ввод элементов одномерного целочисленного массива и вывод введенных значений в одно окно вывода;
- в) ввод элементов одномерного вещественного массива и вывод введенных значений в одно окно вывода.

8. Нижеприведенный фрагмент программы решает следующую задачу:

S = 0

For I = 1 To N Step 2

If A(I) < 0 Then

S = S + A(I)

End If

Next I

- а) вычисление суммы положительных элементов одномерного массива, стоящих на нечетных местах;
- б) вычисление суммы отрицательных элементов одномерного массива, стоящих на четных местах;
- в) вычисление суммы отрицательных элементов одномерного массива, стоящих на нечетных местах.

9. Нижеприведенный фрагмент программы решает следующую задачу:

For I = 1 To M

S = 0

For J = 1 To N

If A(I, J) > 0 Then

S = S + A(I, J)

End If

Next J

A(I, 2) = S

Next I

- а) замена в матрице элементов второй строки на суммы положительных элементов соответствующих столбцов;
- б) замена в матрице элементов второго столбца на суммы положительных элементов соответствующих строк;
- в) замена в матрице элементов второго столбца на суммы отрицательных элементов соответствующих строк.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

- Информатика** : учеб. для вузов / под ред. Н. В. Макаровой. – М. : Финансы и статистика, 2003. – 768 с.
- Гарнаев, А. Ю.** Excel, VBA, Internet в экономике и финансах / А. Ю. Гарнаев. – СПб. : БХВ-Санкт-Петербург, 2005. – 816 с.
- Гарнаев, А. Ю.** Самоучитель VBA / А. Ю. Гарнаев. – СПб. : БХВ-Санкт-Петербург, 2003. – 512 с.
- Гарнаев, А. Ю.** Использование MS Excel и VBA в экономике и финансах / А. Гарнаев. – СПб. : БХВ-Санкт-Петербург, 1999. – 336 с.
- Джелен, Б.** Применение VBA и макросов в Microsoft Excel : пер. с англ. / Б. Джелен, Т. Сирстад. – М. : Вильямс, 2006. – 624 с.
- Олбрайт, К.** Моделирование с помощью Microsoft Excel и VBA: разработка систем поддержки принятия решений : пер. с англ. / К. Олбрайт. – М. : Вильямс, 2005. – 672 с.+ 1 электрон. опт. диск (CD-ROM).
- Уокенбах, Д.** Подробное руководство по созданию формул в Excel 2003 : пер. с англ. / Д. Уокенбах. – М. : Вильямс, 2005. – 640 с.
- Прикладная информатика** : учеб. пособие для вузов / А. М. Морозевич [и др.] ; под ред. А. М. Морозевича. – Минск : Выш. шк., 2003. – 335 с.
- Могилев, А. В.** Информатика : учеб. пособие для вузов / А. В. Могилев, Н. И. Пак, Е. К. Хеннер ; под ред. Е. К. Хеннера. – М. : Академия, 1999. – 816 с.
- Меняев, М. Ф.** Информатика и основы программирования : учеб. пособие для вузов / М. Ф. Меняев. – М. : Омега-Л, 2005. – 432 с.
- Основы информатики и вычислительной техники.** Информатика : практикум к лабораторным работам по одноименным курсам для студентов экономических специальностей дневной и заочной форм обучения / Н. В. Водополова, В. И. Мисюткин, С. А. Чабуркина. – Гомель : ГГТУ им. П. О. Сухого, 2005. – 33 с.
- Основные приемы программирования в объектно ориентированных языках** : практикум по выполнению лабораторных и контрольных работ по курсам «Информатика» и «Основы информатики и вычислительной техники» для студентов экономических специальностей / Н. В. Водополова, В. И. Мисюткин, С. А. Чабуркина. – Гомель : ГГТУ им. П. О. Сухого, 2006. – 41 с.

СОДЕРЖАНИЕ

Пояснительная записка	4
Теоретические основы курса	4
1. Краткие теоретические сведения о создании и использовании макросов.....	4
1.1. Макрос: основные понятия	4
1.2. Запись макроса.....	5
1.3. Удаление макроса	6
1.4. Редактирование макроса	7
1.5. Переименование макроса.....	7
1.6. Вызов макроса или запуск макроса на выполнение	7
1.7. Назначение макросу клавиш быстрого реагирования.....	7
1.8. Назначение макроса графическому объекту	7
1.9. Назначение макроса кнопке	7
1.10. Создание панели управления пользователя	9
2. Пользовательские функции в табличном процессоре Excel.....	11
2.1. Правила набора пользовательских функций.....	11
2.2. Математические пользовательские функции.....	11
2.3. Разветвляющиеся пользовательские функции.....	12
3. Основы программирования в среде VBA.....	12
3.1. Последовательность набора программ в среде VBA	12
3.2. Структура простой программы	14

3.3. Допустимые имена (идентификаторы).....	14
3.4. Типы данных	14
3.5. Описание переменных	15
3.6. Операции, операторы и выражения VBA.....	15
3.7. Оператор присваивания	16
3.8. Правила оформления кода	16
3.9. Общие сведения о вводе-выводе данных	17
3.10. Использование окна ввода	17
3.11. Использование окна сообщений	17
3.12. Обмен информацией с рабочим листом Excel	21
4. Технологии программирования простых циклических алгоритмов в среде VBA.....	22
4.1. Операторы цикла	22
4.2. Табулирование функции	22
4.3. Описание числовых массивов	24
4.4. Ввод и вывод элементов массива.....	24
4.5. Простые операции над одномерными массивами	25
4.6. Определение минимального и максимального элементов массива	26
4.7. Формирование массива	27
5. Технологии программирования вложенных циклов	28
5.1. Вложенные циклы	28
5.2. Двухмерные массивы и их описание	28
5.3. Ввод и вывод двухмерных массивов	29
5.4. Алгоритмы обработки двухмерных массивов	29
6. Технологии обработки символьных данных.....	31
7. Технологии работы с объектами класса «Форма».....	35
7.1. Объект «Форма»: свойства и методы.	35
7.2. Элементы управления объекта «Форма»: их свойства и методы	36
7.3. Обработка событий при работе с объектом «Форма».....	37
7.4. Использование форм для организации диалога с пользователем	38
Задания лабораторных работ.....	40
Лабораторная работа 1	40
Лабораторная работа 2	43
Лабораторная работа 3	44
Лабораторная работа 4	45
Лабораторная работа 5	46
Лабораторная работа 6	49
Лабораторная работа 7	52
Лабораторная работа 8	52
Лабораторная работа 9	54
Лабораторная работа 10	54
Лабораторная работа 11	55
Лабораторная работа 12	57
Лабораторная работа 13	57
Лабораторная работа 14	58
Лабораторная работа 15	58
Лабораторная работа 16	59
Лабораторная работа 17	60
Лабораторная работа 18	61
Лабораторная работа 19	61
Лабораторная работа 20	61
Вопросы для подготовки к тестированию и экзамену по курсу «Технологии программирования» для студентов заочной формы обучения.....	61
Примерные тестовые задания.....	62
Список рекомендуемой литературы	65

Учебное издание

Мовшович Семен Михайлович
Мисюткин Виктор Иванович

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Пособие
для студентов специальности 1-26 03 01
«Управление информационными ресурсами»

Редактор Е. Г. Привалова
Технический редактор И. А. Козлова
Компьютерная верстка Л. Ф. Кириленкова

Подписано в печать 29.08.08. Бумага типографская № 1.
Формат 60 × 84 ¹/₁₆. Гарнитура Таймс. Ризография.
Усл. печ. л. 6,74. Уч.-изд. л. 7,0. Тираж 120 экз.
Заказ №

Учреждение образования
«Белорусский торгово-экономический университет
потребительской кооперации».
246029, г. Гомель, просп. Октября, 50.
ЛИ № 02330/0056814 от 02.03.2004 г.

Отпечатано в учреждении образования
«Белорусский торгово-экономический университет
потребительской кооперации».
246029, г. Гомель, просп. Октября, 50.